FEDERAL UNIVERSITY OF SANTA CATARINA
POSTGRADUATE PROGRAM IN AUTOMATION AND SYSTEM ENGINEERING

Ali Sayyed

EXPLOITING AND OPTIMIZING MOBILITY IN WIRELESS SENSOR NETWORKS

Florianópolis

2016

Ali Sayyed

EXPLOITING AND OPTIMIZING MOBILITY IN WIRELESS SENSOR NETWORKS

Thesis submitted to the Postgraduate Program of the Federal University of Santa Catarina in partial fulfillment of the requirements for obtaining the degree of Doctor in Automation and System Engineering.

Advisor: Prof. Dr. Leandro Buss Becker
Co-Advisor: Prof. Dr. Gustavo Medeiros de Araujo

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Sayyed, Ali
Exploiting and Optimizing Mobility in Wireless Sensor Networks / Ali Sayyed ; orientador, Leandro Buss Becker ; coorientador, Gustavo Medeiros de Araujo. - Florianópolis, SC, 2016.
145 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Inclui referências

1. Engenharia de Automação e Sistemas. 2. Wireless Sensor Network. 3. Mobility. 4. Data Colection. 5. Localization. I. Buss Becker, Leandro. II. Medeiros de Araujo, Gustavo. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Ali Sayyed

EXPLOITING AND OPTIMIZING MOBILITY IN WIRELESS SENSOR NETWORKS

This Thesis has been evaluated and approved for obtaining the degree of **Doctor in Automation and Systems Engineering**, and was also approved in its final format by the Postgraduate Program of Automation and Systems Engineering at the Federal University of Santa Catarina.

Florianópolis, 11 de November 2016.

Prof. Dr. Daniel Ferreira Coutinho
Coordinator
Postgraduate Program in Automation and System Engineering,
Federal University of Santa Catarina, Florianópolis, Brazil

Examination Committee:

Prof. Dr. Leandro Buss Becker
Advisor

Prof. Dr. Gustavo Medeiros de Araujo
Co-Advisor

Prof. Dr. Carlos Eduardo Pereira
Federal University of Rio Grande do Sul, Porto Alegre, Brazil
Presidente

Prof. Dr. Flávio Morais de Assis Silva
Federal University of Bahia, Salvador, Brazil

Prof. Dr. Francisco Vasques
FEUP, University of Porto, Porto, Portugal

Prof. Dr. Marcelo Maia Sobral
Federal Institute of Santa Catarina, Florianópolis, Brazil

Prof. Dr. Joni da Silva Fraga
Federal University of Santa Catarina, Florianópolis, Brazil

Dedicated to my beloved Father and my dearest Mother for their endless and unconditional love. They will hold a very special place in my heart, forever.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious, the Most Merciful. Thank You Allah for the wisdom, strengths, and blessings you give me each day and for all the people around me who make my life easier and more meaningful.

First and foremost, I would like to express my greatest gratitude and sincere thanks to my research advisor Prof. Dr. Leandro Buss Becker for his guidance, support, encouragements, and patience. His deep insight brought me a lot of ideas that were extremely useful for all of my work. I am thankful to him for believing in me and for accepting me as his graduate student despite my limited research experience.

Special thanks goes to Prof^{a}. Dr^{a}. Maria de Lourdes de Souza from PEN/UFSC and Prof. Dr. Jean-Marie Farines and Prof. Dr. Jomi Fred Hübner from DAS/UFSC for their cooperation and support in my Ph.D. admission here.

I am also very grateful to Prof. Dr. Joni da Silva Fraga, Marcelo Ricardo Stemmer and all Professors of the Postgraduate program of Automation and System Engineering at UFSC and the Secretary of PPGEAS, Enio Snoeijer.

I am thankful to Prof. Dr. Gustavo Medeiros de Araujo for his valuable comments and feedback in improving my research work.

I would also acknowledge the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Departamento de Automação e Sistemas (DAS/UFSC) for the Doctoral Research Fellowship.

Last but not least, my deepest gratitude goes to my beloved parents (Mian Sayyed Qalandar and Mehir Bano) for their endless love, prayers, support and encouragement. I wish to express my deepest gratitude to my sisters and brothers for giving me strength to reach for the stars and chase my dreams. I am thankful to my wife (Najma Naz) for loving, enduring and supporting me and helping me out in all difficult times. I am very grateful to all my friends, teachers, and mentors since my childhood, I think without being trained and taught well, I would not have been able to complete my higher studies. I am thankful to those who indirectly contributed to this research, your kindness and love mean a lot to me. Thank you very much.

Thank you, Lord, for always being there for me. This thesis is only a beginning of a journey

Florianopolis Brazil.
Friday, August 15, 2016

Papers included in this Thesis

A book chapter (SAYYED; BECKER, 2015b) out of the preliminary literature review for this thesis has been published as ”A Survey on Data Collection in Mobile Wireless Sensor Networks” by Ali Sayyed, Leandro Buss Becker in *Cooperative Robots and Sensor Networks 2015*, Springer International Publishing.

A paper (SAYYED et al., 2015) out of the preliminary research work carried out for this thesis has been published as ”Dual-stack single-radio communication architecture for UAV acting as a mobile node to collect data in WSNs” by Ali Sayyed, Gustavo Medeiros de Araújo, João Paulo Bodanese and Leandro Buss Becker in *MDPI Sensors.*

Chapter 2, in part, has been accepted for publication as ”EGL: Efficient Geometry-based Localization in Wireless Sensor Networks” by Ali Sayyed, Gustavo Medeiros de Araujo, Leandro Buss Becker in *International Journal of Sensor Networks*, Inderscience Publishers.

Chapter 3, in part, has been published in (SAYYED; BECKER, 2015a) as ”Optimizing speed of mobile data collector in Wireless Sensor Network” by Ali Sayyed, Leandro Buss Becker in *2015 IEEE International Conference on Emerging Technologies (ICET)*, Peshawar Pakistan.

Chapter 3, in part, has also been published in (SAYYED; ARAÚJO; BECKER, 2016) as ”Smart Data Collection in Large Scale Sparse WSNs” by Ali Sayyed, Gustavo Medeiros de Araujo, Leandro Buss Becker in *2016, 9th IFIP Wireless and Mobile Networking Conference*, Colmar France.

Extended versions of chapter 3 and a paper out of Chapter 4 will be submitted to prestigious journals soon.

”Education is the most powerful weapon which you can use to change the world” (Nelson Mandela)

RESUMO EXTENDIDO

Nos últimos anos, as chamadas Redes de Sensores Sem Fio (RSSF) tem sido usadas numa grande variedade de aplicações, tais como monitoramento (p.ex. poluição do ar e água, vulcões, estruturas, sinais vitais), detecção de eventos (p.ex. vigilância, incêndios, inundações, terremotos), e monitoramento de alvos (p.ex. segurança, animais silvestres, etc). RSSF são constituídas tipicamente por dezenas, as vez centenas de pequenos dispositivos alimentados por baterias, capazes de realizar medições e de transmitir tais dados para uma estação base através de um canal sem fio.

Uma das formas mais promissoras para melhorar o desempenho das RSSF em termos de conectividade, tempo de vida da rede, e latência na transmissão dos dados é através de técnicas que exploram a mobilidade em um ou mais componentes da rede. A mobilidade na RSSF pode ser tanto controlável como aleatória, sendo que em ambos os casos os protocolos devem ser devidamente ajustados para responder adequadamente aos cenários em questão. No caso de mobilidade aleatória, os nodos sensores podem ser capazes de aprender os padrões de mobilidade dos nodos para poderem otimizar a operação da rede. Por outro lado, sendo os padrões de mobilidade conhecidos, é possível fazer escolhas para melhor sintonizar o desempenho da rede de acordo com os critérios estabelecidos pelo projetista.

A presente tese de doutorado procura explorar as vantagens associadas com o uso de mobilidade controlada em RSSF. É possível definir mobilidade controlada como sendo a capacidade de se alterar propositalmente o posicionamento de determinados nodos da RSSF. Com isso se torna possível explorar, controlar, ou mesmo otimizar a trajetória e a velocidade dos nodos móveis da RSSF a fim de maximizar o desempenho da rede como um todo. Definitivamente, o uso de nodos que permitam o ajuste de trajetória e velocidade oferece um alto grau de flexibilidade para se explorar aspectos de mobilidade e projetar protocolos de coleta de dados otimizados.

Ao se utilizar mobilidade controlada, algumas das operações realizadas pela RSSF podem ser significativamente melhoradas, de modo a tornar possível ajustar o padrão de desempenho da rede de acordo

com os níveis desejados. Por exemplo, o processo de descoberta de nodos pode ser melhorado e mesmo simplificado com o controle dos nodos móveis, de modo que ele possa se aproximar dos nodos estáticos em instantes pré-determinados. Da mesma forma, o processo de coleta de dados pode ser otimizado se os nodos móveis se moverem mais rapidamente nos locais onde eles precisam coletar menos dados. Entretanto, diversos desafios aparecem neste tipo de contexto. Por exemplo, como se deve escalonar a chegada do(s) nodo(s) móvel(is) e como se deve controlar e otimizar a movimentação em termos de velocidade sem afetar a qualidade de serviço.

Nesse contexto, o segundo capítulo da tese apresenta um esquema de estimação de localização de nodos estáticos espalhados ao longo de uma área predeterminada, utilizando-se para tanto de um nodo móvel com mobilidade controlada. Tal informação de posicionamento é muito importante para a organização de uma RSSF. Com isso é possível definir a sua cobertura, os protocolos de roteamento, a forma de coleta de dados e também auxiliar em aplicações de rastreamento e detecção de eventos. O esquema proposto consiste de uma técnica de localização para estimar a posição dos nodos sensor de forma eficiente, usando apenas um nodo móvel e técnicas geométricas simples. O esquema não requer hardware adicional ou mesmo comunicação entre nodos sensores, evitando assim maiores gastos de baterias. A estimativa de posição obtida é precisa e capaz de tolerar um certo grau de obstáculos. Os resultados obtidos ao longo da tese demostram que a precisão de localização pode ser bem ajustada selecionando corretamente a velocidade, o intervalo de transmissão de beacons e o padrão de varredura da área de interesse pelo nodo móvel.

Já o terceiro capítulo apresentada uma técnica de otimização para fins de controle da mobilidade do nodo coletor de dados (MDC). Com isso torna-se possível desenvolver um esquema inteligente de coleta de dados na RSSF. Em primeiro lugar, são destacados os fatores que afetam o processo de coleta de dados usando um MDC. Em seguida é apresentado um algoritmo adaptativo que permite ajustar os parâmetros de controle necessários para modificar os parâmetros de movimentação do MDC. Estes parâmetros permitem que a velocidade do MDC seja ajustada em tempo de execução para otimizar o processo de coleta de dados. Com isso o MDC pode se adaptar às diferentes taxas de coletas de dados impostas por um conjunto de nodos heterogêneos. O esquema proposto apresenta vantagens significativas para RSSF de grande escala

e também heterogêneas (onde os sensores possuem taxas de amostragem variáveis). Os resultados obtidos mostram um aumento significativo na taxa de coleta de dados e a redução no tempo total de deslocamento e no número de voltas que o MDC gasta para coletar os dados dos sensores.

Por fim, o capítulo 4 propõe um mecanismo de controle de acesso (MAC) adaptado ao cenário de mobilidade, que se ajusta automaticamente de acordo com o padrão de mobilidade do MDC. O mesmo foca uma redução no consumo de energia e na melhoria da coleta de dados, suportando mobilidade e evitando colisões de mensagens. Este protocolo destina-se a aplicações de coleta de dados nas quais os nós sensores têm de reportar periodicamente a um nó receptor ou estação base. O conceito básico é baseado em acesso múltiplo de divisão de tempo, onde a duração do padrão de sono-vigília é definida de acordo com o padrão de mobilidade do MDC. O esquema proposto é capaz de atender tanto mobilidade aleatória quanto controlada por parte do MDC, desde que as RSSF sejam organizadas em cluster. Uma análise de simulação detalhada é realizada para avaliar seu desempenho em cenários mais gerais e sob diferentes condições operacionais. Os resultados obtidos mostram que o nosso esquema proposto supera amplamente o protocolo 802.15.4 com sinais (beacons) em termos de eficiência energética, tempo de deslocamento do MDC e taxas de coleta de dados.

Palavras-chave: RSSF, Mobilidade em RSSF, Cooperação, Controle Dinâmico de Velocidade, Robôs Móveis, Localização, MAC, Protocolos Sleep/Wake-up em RSSF

ABSTRACT

One of the promising techniques for improving the performance of a wireless sensor network (WSN), in terms of connectivity, network lifetime, and data latency, is to introduce and exploit mobility in some of the network components. Mobility in WSN can be either uncontrollable or controllable and needs to be optimized in both cases. In the case of uncontrolled mobility, sensor nodes can learn the mobility patterns of mobile nodes to improve network performance. On the other hand, if the mobility is controllable in terms of trajectory and speed, it can be best tuned to enhance the performance of the network to the desired level. This thesis considers the problem of exploiting and optimizing mobility in wireless sensor networks in order to increase the performance and efficiency of the network.

First, a location estimation scheme is discussed for static nodes within a given sensor area using a controlled mobile node. Position information of static nodes is very important in WSN. It helps in effective coverage, routing, data collection, target tracking, and event detection. The scheme discusses a localization technique for efficient position estimation of the sensor nodes using a mobile node and simple geometric techniques. The scheme does not require extra hardware or data communication and does not make the ordinary sensor nodes to spend energy on any interaction with neighboring nodes. The position estimation is accurate and efficient enough to tolerate obstacles and only requires broadcasting of beacon messages by the mobile node. Obtained simulation results show that the localization accuracy can be well adjusted by properly selecting the speed, beacon interval, and scan pattern of the mobile node.

Second, an optimization technique for controlled mobility of a mobile data collector is presented in order to develop a smart data collection scheme in WSN. In this case, first, the factors affecting the data collection process using an MDC is highlighted. Then, an adaptive algorithm and control parameters that the MDC uses for autonomously controlling its motion is presented. These parameters allow the speed of the MDC to be adjusted at run time in order to adaptively improve the data collection process. Built-in intelligence helps our system adapting to the changing requirements of data collection. Our scheme shows significant advantages for sparsely deployed, large scale sensor networks and heterogeneous networks (where sensors have variable sampling rates). The simulation results show a significant increase in data collection rate and reduction in the overall traverse time and number of laps that the MDC spends for data gathering.

Finally, a mobility aware adaptive medium access control (MAC) is proposed for WSNs which automatically adjusts according to the mobility pattern of the MDC, focusing on reducing energy consumption and improving data collection, while supporting mobility and collision avoidance. This protocol is targeted to data collection applications (e.g. monitoring and surveillance), in which sensor nodes have to periodically report to a sink node. The core concept is based on adaptive time division multiple access, where the sleep-wake duration is defined according to the MDC mobility pattern. The proposed scheme is described for random, predictable, and controlled arrival of MDC in cluster-based WSNs. A detailed simulation analysis is carried out to evaluate its performance in more general scenarios and under different operating conditions. The obtained results show that our scheme largely outperforms the commonly used 802.15.4 beacon-enabled and other fixed duty-cycling schemes in terms of energy efficiency, MDC traverse time, and data collection rates.

Keywords: WSN, Mobility in WSN, Cooperation, Dynamic Speed Control, Mobile Robot, Localization, MAC, Medium Access Control, Sleep/Wake-up in WSN

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Network
MWSN	Mobile Wireless Sensor Network
MDC	Mobile Data Collector
MN	Mobile Node
UAV	Unmanned Aerial Vehicle
CH	Cluster Head
CM	Cluster Members
BS	Base Station
LQ	Link Quality
GPS	Global Positioning System
RSSI	Recieved Signal Strength Indicator
AOA	Angle of Arival
TOA	Time of Arival
EGL	Efficient Geometry-based Localization
MASWS	Mobility Aware Sleep Wake-up Scheduling

CONTENTS

1 INTRODUCTION

In recent years Wireless Sensor Networks (WSNs) have become an established technology for a large number of real-world applications, such as monitoring (e.g., prevention of pollution, agriculture, volcanoes, structures and buildings health), event detection (e.g., intrusions, fire and flood emergencies), and target tracking (e.g., surveillance and monitoring) (AKYILDIZ et al., 2002). WSNs usually consist of hundreds, in some cases, thousands of battery operated tiny devices that measure and collect data from its surrounding environment and forward it to a base station or sink.

Mobility is often introduced in WSN for a number of possible advantages (CHAKRABARTI; SABHARWAL; AAZHANG, 2003a; ANASTASI; CONTI; FRANCESCO, 2009b; LIU et al., 2005a). Mobility can be introduced in any component of a sensor network including the regular sensor nodes, relay nodes (if any), data collectors or sink node or any combination of them. For instance, regular sensor nodes can be mobile. These sensor nodes are the source of information in a sensor network and they perform sensing as their main task. In addition, they may also forward data coming from other nodes which they have been previously in contact with. Similarly, sink or base station, which is the destination or consumer of messages originated by sensors, can also be mobile. These nodes represent the endpoints of collected data in WSNs. Mobile sinks can collect data either directly (i.e., by visiting sensors and collecting data from each of them) or indirectly (i.e., through relays or other nodes) (CHATZIGIANNAKIS; KINALIS; NIKOLETSEAS, 2006). Finally, relay nodes, which are neither producer nor consumer of information in a sensor network, can also be made mobile. Relay nodes perform specific tasks by collecting messages from sensor nodes when in their coverage range, possibly carrying the data to a different location with them and eventually passing it to the base station whenever possible. This significantly reduces not only collisions and message losses, but also minimizes the burden of data forwarding task by nodes and as a result, spreads the energy consumption more uniformly throughout the network (WANG et al., 2005).

Allowing sensor nodes to be mobile increases the number of possible applications beyond the limits of those for which static sensors can be used. Sensors can be attached to people (for monitoring heart rate, blood pressure etc.) (YAN et al., 2010), and to animals for tracking their movements (monitoring migration patterns, feeding habits etc.)

(EHSAN et al., 2012; DYO et al., 2012). Sensors may also be attached to unmanned aerial vehicles (UAVs) for surveillance or environment mapping (WHITE et al., 2008).

The most important characteristic of mobility is controllability. Mobility can be either uncontrollable or controllable. In the case of *uncontrolled mobility*, sensor nodes can learn the mobility patterns of mobile nodes to improve network performance. In this case the more the randomness in mobility, the harder it is to learn and predict. On the other hand in *controlled mobility* the mobile node can actively and deliberately change its trajectory/speed according to the network requirement.

Based on the classification in (KANSAL et al., 2004a), node mobility can be described as one of the four following types:

- **Stationary:** A node which remains static throughout its lifetime.
- **Random:** A node which moves in a random manner following some probabilistic model.
- **Predictable:** A node whose motion and mobility pattern can be known with some accuracy.
- **Controlled:** A node whose motion can be actively changed either autonomously or by a user.

1.1 RESEARCH CONTEXT AND MOTIVATIONS

The present research work focuses on controlled mobility, which can be defined as the ability of mobile nodes to actively change their location in a WSN. Controlling mobility refers to exploiting, controlling, or optimizing the trajectory and speed of mobile nodes in WSNs in order to maximize performance. Clearly, a controlled mobile node gives more flexibility in exploiting mobility and designing data collection protocols for mobile wireless sensor networks (MWSNs).

In the presence of controlled mobility, some of the WSN operations can be significantly improved, so that the performance of the sensor network can be enhanced to a desired level (FRANCESCO; DAS; ANASTASI, 2011). For example, the discovery process of mobile nodes can be improved and simplified by controlling the mobile nodes so that it visits the static nodes at specific times. Similarly, data collection process can be enhanced if the mobile nodes move slower or even stop at each static node until finishing the collection of all buffered data.

However, different challenges may arise in this context. For example, how to schedule the arrival of mobile nodes and how to control and optimize the mobility in terms of trajectory and speed, while at the same time maintaining the quality of service (QoS) constraints.

1.1.1 Roles of Mobile Nodes in WSNs

Mobility may exist in a sensor network in the following forms.

Mobile Sensors. Mobility may exist in ordinary or regular sensor nodes which are the sources or origin of information in WSNs. In addition, these nodes may also forward or relay messages in the network. For instance, in (YAN et al., 2010; EHSAN et al., 2012; DYO et al., 2012; CHOI et al., 2008) animals/people with attached sensors, not only generate their own data but also carry and forward data coming from other nodes which they have been previously in contact with. They eventually transfer all their data when in contact with the sink or base station.

Mobile Sinks. Mobile Sink (or Mobile Base Station) refers to mobile nodes which are the destination or consumer of messages originated by sensors. Mobile sinks collect data sensed by sensor nodes either directly (i.e., by visiting sensors and collecting data from each of them) or indirectly (i.e., through relays or other nodes). For instance, a mobile sink is used in (CHATZIGIANNAKIS; KINALIS; NIKOLETSEAS, 2006) and (WANG et al., 2005) to move in the network area and collect data from sensors.

Mobile Relays. Relay nodes are neither producer nor consumer of messages in a sensor network. They perform a specific task by collecting data from sensor nodes when in their coverage range, carry the data to a different location with themselves and eventually pass it to the base station. Data collection using mobile relays has been proposed in (SHAH et al., 2003a) where the network is based on a three-tiered architecture, with the middle tier being represented by mobile relays.

Mobile Data Collector. Mobile data collector is basically a mobile relay or a mobile sink node. It moves into different areas of the network, collects messages, and either consumes it itself or eventually passes it to the base station whenever possible.

1.1.2 Advantages of Mobility in WSNs

- *Coverage and Connectivity.* In the presence of mobility, dense (re)deployment is not necessary for sensor networks. In this case, mobile nodes can cope with isolated regions and cover the holes created in the connectivity of network due to dead nodes or sparseness (LIU et al., 2005a).
- *Reliability.* Mobile nodes can move to different regions of the network and collect data directly through single-hop transmission, thereby reduces the number of collisions and message losses and as a result, increase the probability of successful transmissions (ANASTASI; CONTI; FRANCESCO, 2009b).
- *Lifetime.* In WSNs, nodes near to the sink deplete their energy much faster than the other nodes because they sense and forward their own data as well as data of other nodes which are far away from the sink. Mobile nodes can disperse the energy consumption and transmission more uniformly as shown in (GANDHAM et al., 2003) and (WANG et al., 2005).
- *Target Tracking.* In many real world applications of *object tracking*, sufficient sensor nodes need to be deployed along the track of the target. In addition, more expensive sensing devices, e.g. camera, should be required to get more information. Nevertheless, it is infeasible to deploy a large number of sensors and at the same time equip each one with a camera to tackle the situation. Controlled mobility in MWSNs can be very helpful in these type of applications as shown in (LAMBROU; PANAYIOTOU, 2007) and (VERMA; SAWANT; TAN, 2006a).
- *Channel Capacity.* Experiments have shown that exploiting mobility gives us greater channel capacity and data integrity due to multiple communication pathways, and less number of hops for data delivery and dissemination (KANSAL et al., 2004a).

1.1.3 Challenges of Mobility in WSNs

Mobility in WSNs also introduces significant challenges, described as follows.

- *Scheduling.* Determining when an activity (detection, communication etc.) with mobile node should start/end and for what duration and with what resource is always a challenging task in MWSNs and especially when sensor nodes are sampling at different rates, in which case some nodes need to be visited more frequently than others (SOMASUNDARA; RAMAMOORTHY; SRIVASTAVA, 2004a).

- *Reliability.* Reliability can also be a challenge in MWSNs because the time available for detection and communication with a mobile node is scarce and short due to its movement. Paths may break frequently due to channel fading, interference, and node mobility.

- *Mobility.* Mobility in MWSNs can be either uncontrollable or controllable and need to be optimized in both cases (FRANCESCO; DAS; ANASTASI, 2011). In the former case, mobility patterns of mobile nodes can be learned and predicted to enhance detection and transmission process. In the latter case, the trajectory and speed of mobile nodes can be optimized in order to increase network performance. For instance, mobile nodes can be programmed to visit static nodes at specific times, and at the same time can slow down or even stop while close to nodes until all buffered data is collected.

- *Localization.* Node localization is one of the most significant challenges in WSNs (AMUNDSON; KOUTSOUKOS, 2009). In static WSN, node positions can be calculated during the initialization stage. However, those nodes that are mobile must continuously obtain their position as they keep moving.

- *Dynamic Network Topology.* Due to dynamic network topology, new routing, MAC, and scheduling protocols are needed to optimize the performance in MWSNs. For instance, static WSN routing protocols can provide the required functionality but cannot handle mobility, whereas, Mobile Ad Hoc Network (MANET) routing protocols can deal with mobility in the network but they are not designed for one-way communication, which is often the case in sensor networks (LAMBROU; PANAYIOTOU, 2009). In addition, MWSNs differ from MANET in many ways. For example in the number of nodes (density of deployment), energy requirement and traffic requirement (MWSNs are highly data driven).

1.2 THESIS CONTRIBUTIONS

This thesis investigates the exploitation and optimization of controlled mobility in WSNs in order to increase the performance and efficiency of the network.

First, the localization of static sensor nodes using a controlled mobile node is discussed. Position information of static nodes is very important in WSN. It helps in effective coverage, routing, data collection, target tracking, and event detection. The localization scheme describes a technique for efficient position estimation of the sensor nodes using a controlled mobile node and simple geometric techniques. The scheme does not require extra hardware or data communication and does not require the static sensor nodes to spend energy on any interaction with their neighboring nodes. This scheme is accurate and efficient enough to tolerate obstacles and only requires broadcasting of beacon messages by the mobile node. Obtained simulation results show that localization accuracy can be kept low and well adjusted by properly selecting the speed, beacon interval, and scan pattern of the mobile node.

Second, a framework for optimizing the motion of mobile data collector is formulated. In this case, the elements affecting the data collection process using an MDC is discussed. Also, an adaptive algorithm (along with its control parameters) is proposed so that the MDC can autonomously control its motion. These parameters allow the speed of the MDC to be adjusted at run time in order to adaptively improve the data collection process. Built-in intelligence helps the system adapting to the changing requirements of data collection. The proposed scheme shows significant advantages for sparsely deployed, large scale sensor networks and heterogeneous networks (where sensors have variable sampling rates). The simulation results show a significant increase in data collection rate and reduction in the overall traverse time and number of laps that the MDC spends for data gathering.

Finally, a mobility aware adaptive medium access control for WSN with MDC is proposed. The core concept is based on adaptive time division multiple access where the sleep-wake duration is defined according to the mobility pattern of MDC. The protocol is discussed for an MDC with random, predictable, and controlled mobility in a cluster-based WSNs. A detailed simulation analysis is carried out to evaluate its performance in more general scenarios and under different operating conditions. The obtained results show that the proposed scheme largely outperforms commonly used fixed duty-cycling schemes in terms of energy efficiency, MDC traverse time, and data collection rates.

1.3 OBJECTIVES AND SCOPE LIMITS

The overall objective of this work is the study of how to exploit and optimize controlled mobility in wireless sensor networks in order to increase the performance and efficiency of the network. This general goal is further divided into two more specific goals:

- The first objective was to efficiently and accurately localize static sensor nodes using a controlled mobile node in WSN with minimum possible localization error. The goal was to provide a solution that uses simple geometric techniques without any extra hardware or data communication overhead and which does not require the ordinary sensor nodes to spend energy on any interaction with their neighboring nodes.
- The second objective was to provide a strategy and develop an adaptive algorithm and control parameters that an MDC uses for autonomously controlling its motion during data collection process in sparse and large-scale sensor networks. The goal was to develop a scheme that allows the speed of the MDC to be adjusted at run time in order to adaptively improve the data collection rates and minimize traverse time of MDC. Minimizing the traverse time of MDC does not only reduce the expenses related to MDC movement but also minimize data transfer delays. Dynamic speed control is also associated with reduced buffer overflows, and increased data transfer rates and network lifetime, as shown for low-speed ground robots in (CHATZIGIANNAKIS; KINALIS; NIKOLETSEAS, 2006).
- The third objective was to develop a mobility aware adaptive medium access control for WSN which is based on an adaptive time division multiple access where the sleep-wake duration is defined according to the mobility pattern of MDC. The goal was to achieve low power consumption, minimize collisions, and improve data collection rates.

It must be noted that this work does not focus on the study of problems like sensor data fusion or aggregation. Initial path/trajectory planning for MDC for localization and data collection (with given network area and a number of clusters/nodes) is out of the scope of this work.

1.4 DOCUMENT ORGANIZATION

The remaining parts of this document are organized as follows. Chapter 2 discusses the localization estimation scheme, chapter 3 presents the adaptive speed control, and chapter 4 focuses on the adaptive medium access control. Finally, chapter 5 summarizes the conclusions and future research direction. Each of chapters 2, 3, and 4 first discuss the related works, followed by the presentation of the proposed approach and its evaluation.

2 LOCALIZATION ESTIMATION

2.1 INTRODUCTION

As already presented, WSNs consist of spatially distributed sensors devices used for a variety of different applications ranging from environmental monitoring to event detection, and from military applications to industrial automation.

One way of deploying a WSN is to randomly scatter the static nodes in the region of interest which make the resultant network topology random. Such deployments are often being used for networks intended to perform a number of tasks, ranging from environmental and natural habitat monitoring to smart battlefields and health applications (PAL, 2010). However, most real-world applications depend on the location information of sensor nodes in some fixed coordinate system. Localization is one of the most important issues in a sensor network since position information of nodes are very useful and offer a number of advantages in efficient coverage, routing, data collection, target tracking, and event detection etc (HIGHTOWER; BORRIELLO, 2001). For instance, in the case of an event detection or node failure, position information enables a remote user about the precise location of the event or failure within a network. Similarly, positions of sensor nodes also help in building efficient routing paths to the base station. Unfortunately, for a variety of sensor networks, the straightforward solution of adding a GPS unit to all nodes in the network is often not feasible. It is, therefore, important to design cost-effective, rapidly deployable, and efficient localization schemes in such environments.

Mobile elements are often introduced in WSNs for a number of possible advantages including data collection (SAYYED et al., 2015; SAYYED; BECKER, 2015a), coverage and connectivity (LIU et al., 2005b), Target Tracking (VERMA; SAWANT; TAN, 2006b), event detection (LAMBROU; PANAYIOTOU, 2007), and localization (CHEN et al., 2010; MUNIR; BIN; JIAN, 2007; SSU; OU; JIAU, 2005).

This chapter discusses a geometry-based localization scheme for static sensor nodes in WSN. A mobile node (MN) is used to periodically broadcast its current location information while traversing the sensing field and assist in finding the position of stationary nodes spread along an area of interest. The adopted technique is based on the fact that whenever the **radius** of a circle and **three boundary points** on that circle is known, the center of the circle (the location of sensor node) can be estimated. The scheme does not require extra hardware or data communication and does not require the ordinary sensor nodes to spend energy on neighboring interaction for localization. In this method, first, a mobile node is used to scan the area of interest with a predefined pattern according to the sensor field. The trajectory of the scan pattern of the mobile node is designed so that it visits all sensor nodes in the field at least twice. The mobile node periodically broadcasts its position information to all sensor nodes within the sensor field. The sensor nodes make use of the MN broadcasts and location information in order to estimate their own positions.

There are several factors that influence the quality of the position calculations, such as the speed, scanning pattern, and beacon transmission frequency of the mobile node. Besides, there are techniques used for detecting and correcting errors in the selection of boundary points. This thesis presents results that show the extent to which these factors influence the quality of the position estimations.

The reminder parts of this chapter are organized as follows: Section 2.2 describes some of the related works in the field of localization, including range-based, range-free, and mobile beacon based techniques. An overview of the proposed scheme along with the adopted localization principles, enhancements, and a working algorithm is discussed in section 2.3. Simulation results and performance analysis are presented in section 2.4.

2.2 RELATED WORKS

In WSN, accurate and low-cost sensor localization is considered important in a wide range of applications (PATWARI et al., 2005). Detailed reviews of location estimation algorithms have been presented in (BACHRACH; TAYLOR, 2005; PAL, 2010). The performance of a localization technique mainly depends on the incorporated algorithm used and size, density and environmental condition of the sensor network (PATWARI et al., 2005). Over the years, several techniques have been proposed and tested to estimate the location of nodes in a sensor network. Localization techniques can be broadly classified into the following categories.

2.2.1 Range-free Localization

In range-free localization, network constraints such as connectivity are used to estimate the position of nodes instead of distance or angle. For instance, in (CORKE; PETERSON; RUS, 2007) static nodes use GPS data from mobile nodes and simple averaging techniques on received signal strength to estimate their locations. Similarly in (CHEN et al., 2010), considering obstacles in sensor area, a mobile node cooperates with static nodes and moves actively to refine its location. The scheme uses a relay node and a novel convex position estimation algorithm to effectively increase accuracy and decrease the effects of obstacles on node localization.

Belusu and Heidemann used Centroid formula to calculate nodes' positions using reference points positions from received beacons (BULUSU; HEIDEMANN; ESTRIN, 2000). Centroid scheme was further improved in (BULUSU; HEIDEMANN; ESTRIN, 2001) with adaptive beacon placement algorithm. Niculescu and Nath proposed DV-Hop localization method (NICULESCU; NATH, 2003), which approximate the location of sensor nodes by measuring hop counts from each node to specific anchor points and triangulation. The DV-Hop scheme was further improved using RSSI technology in (RABAEY; LANGENDOEN, 2002).

A range-free localization scheme for wireless sensor networks using mobile nodes equipped with a GPS and four directional antennas has been proposed in (OU, 2011). Each mobile node broadcasts its position information as it moves through the WSN. The sensor nodes receive these messages and utilize a simple processing scheme to determine their own location. The scheme does not require any ranging hardware or communications between the sensor nodes.

In the work presented in (LEE; CHOI; KIM, 2013), the authors proposed a multidimensional support vector regression (MSVR) to enhance the localization accuracy. This scheme is based on a new MSVR training method that allows multiple outputs and localizes the sensors without using multilateration. The work requires the existence of multiple anchors nodes evenly distributed to increase the reliability of the localization determination.

Similarly, authors in (GUERRERO; ALVAREZ; RIVERO, 2010) propose a three-dimensional range-free localization algorithm for WSNs, using an aerial mobile node equipped with a rotary and tilting directional antenna. The algorithm runs locally on each sensor node and is based only on the analysis of the information received from the mobile node.

Some range-free localization techniques, such as (HE et al., 2003a), require extensive communication between neighboring sensor nodes. Hence, localization accuracy and communication overhead are the most critical factors in range-free approaches. Interested readers should refer to (HE et al., 2003a) for a detailed survey on range-free localization.

2.2.2 Range-based Localization

Range-based schemes somehow measure the range information (*distance* or *angle*) between each node in order to find its location. The ranging information can be obtained using a number of different techniques. For instance, RSS (ARIAS et al., 2004), TOA (VENKATRAMAN; CAFFERY; YOU, 2004), AOA (NICULESCU; NATH, 2001) etc.

- *Received Signal Strength Indicator(RSSI).* It is defined as the measured voltage or power (estimated from signal strength) by a receiver node. RSS measurements are simple and inexpensive and can be done by each node receiver during normal data communication without using additional resources but can be unpredictable (PATWARI et al., 2005). An RSSI based distance measurement model which estimate the distance between sensor nodes in WSNs is presented in (ADEWUMI; DJOUANI; KURIEN, 2013). The scheme describes the relationship between the received power, transmitted power and the distance among wireless sensor nodes. Similarly, a location map of the network nodes is built in (GU; CHEN; SUN, 2011), using the partial data pair, RSSI, and physical location. The authors introduced a mechanism called as partially paired locality correlation analysis (PPLCA). The PPLCA handles with the semi-paired scenario of wireless sensor network localization by the combination of the neighborhood structure information in data.

- *Time of Arrival (TOA).* It is the time at which a signal first arrives at a receiver node. It is the time of transmission plus propagation delay and is equal to the transmitter-receiver distance divided by the propagation velocity. Arrival time for the line-of-sight signal can be easily measured by receivers, but this calculation is affected by additive noise and multi-path signals (PATWARI et al., 2005).

- *Angle of Arrival (AOA).* It is the direction of arriving signals to neighboring sensors (PATWARI et al., 2005). The most common method to find the angle of arrival is to use an array of antennas and hence array signal processing at receiver nodes. The AOA is estimated from the difference in arrival times for a transmitted signal at each of the sensor array elements. The major drawback of this technique is the need for multiple antenna elements, which increase device cost and size.

Range-based schemes are typically accurate, having less than 5 m location error, but they require more hardware on sensor nodes (HIGHTOWER; BORRIELLO, 2001).

2.2.3 Centralized Localization

Centralized localization require the migration of inter-node ranging and connectivity data to a sufficiently powerful centralized node and then transferring back the resulting locations to respective nodes. The main advantage of centralized algorithms is that they avoid the problem of computation locally on each node. However, their main drawback is the communication overhead of moving data back to a centralized node and transferring it back to sensor nodes. Works presented in (SHANG et al., 2003; KANNAN; MAO; VUCETIC, 2006; ALIPPI; VANINI, 2006) can be regarded as the representative proposals in this category.

2.2.4 Distributed Localization

In distributed localization algorithms, all the relevant computations for location estimation are done locally on the sensor nodes. Sensor nodes interact and communicate with each other in order to estimate their positions in a network without the need for a centralized node. However, the main drawback of this technique is the computation overhead, if huge, which is to be done at each node. The works presented in (GUERRERO; ALVAREZ; RIVERO, 2010; SICHITIU; RAMADURAI, 2004; OU, 2011) can be regarded as example proposals in this category.

2.2.5 Mobile Beacon Based Localization

Many schemes have been proposed, using mobile node as a beacon node, in estimating the coordinates of sensor nodes in a wireless sensor network. For instance, a localization mechanism described in (SICHITIU; RAMADURAI, 2004), uses a single mobile beacon transmitting its current location. A sensor node receiving the beacon can approximate its position in an area around the mobile beacon with a certain probability. Together with RSSI measurement, possible locations of the sensor node can be estimated. The more the number of beacons, the more the accuracy of this scheme. The precision of the scheme is good as long as each sensor node receives at least three beacon messages.

Mobile beacon together with ranging information has also been used by Sun and Guo in a probabilistic localization scheme (SUN; GUO, 2004). In this case, TOA techniques together with Centroid formula with distance information are used to calculate the position of a sensor node. Galstyan et al. proposed a range-free localization scheme to minimize the uncertainty in nodes positions using radio connectivity constraints (GALSTYAN et al., 2004). Location of receiver node is bounded in the transmission area of the sender with each beacon message.

Mobile anchor points are used in (SSU; OU; JIAU, 2005), as reference points for estimating the location of static nodes in a wireless sensor network. The scheme develops a localization mechanism using mobile anchor points and a geometry conjecture. The conjecture is based on the fact that a perpendicular bisector of a chord passes through the center of the circle. The same technique of geometry is used in (SINGH; KHILAR, 2016), where the intersection point of two perpendicular bisectors of the chords is taken as the estimated location of the sensor node. However, in this case, to determine three beacon points for establishing two chords, the sensor node assumes a circle which passes through a beacon point that is farthest among other (by RSSI). The reminder two beacon points are then approximated by the geometry of an arc.

A localization scheme based on the conjecture of perpendicular bisector of chord has been proposed in (OU; HE, 2013), where the sensor nodes are helped by mobile nodes for estimating their locations. However, the proposed method attempts to optimize the trajectory of the mobile anchor node in order to minimize the localization error and to guarantee that all sensor nodes can determine their locations.

Another mobile anchor node localization algorithm is proposed in (HAN et al., 2014). It is based on regular hexagon (MAALRH) in two-dimensional WSNs. In spite of a good accuracy, the proposal assumption is based on several mobile nodes equipped with GPS units moving in the sensing field. The localization algorithm is based on the idea of trilateration and regular hexagon, which can cover the whole sensing area with a boundary compensation method.

Similarly, another scheme proposed in (MUNIR; BIN; JIAN, 2007) makes use of the communication range of the sensor nodes and motion of a mobile sink in a straight line. Based on the communication range of each sensor node, a geometrical topology representation of the sensor nodes is formed by mobile sink and then the inter-node distance is calculated by mobile sink with reference to its line of motion.

2.3 EFFICIENT GEOMETRY-BASED LOCALIZATION

The proposed *Efficient Geometry-based Localization* (EGL) aims to provide a flexible, scalable, and distributed localization scheme for WSNs using a single mobile node.

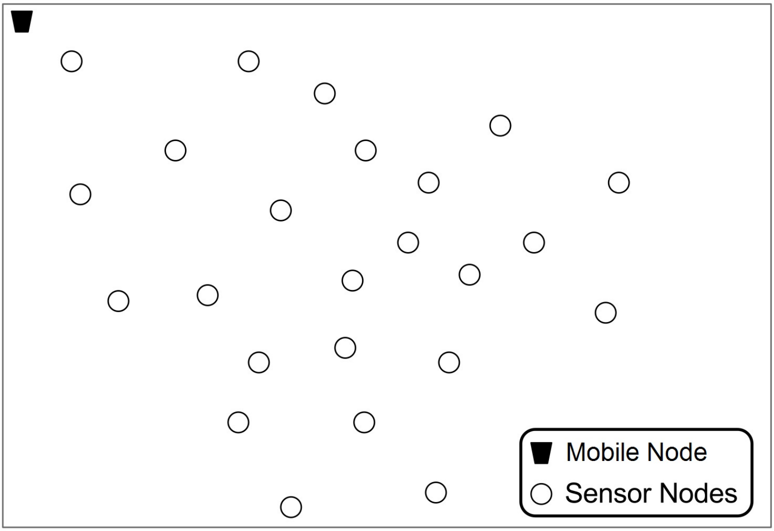


Figure 1 – WSN Scenario for Geometry-based Localization

Figure 1 illustrates a scenario where several static sensor nodes are deployed randomly in the sensing field. A mobile node (Aerial or Ground) is used to scan the network area and help sensor nodes to determine their locations. The mobile node periodically broadcast beacon messages while traversing the network area. Sensor nodes, which are in the range of beacon messages, estimate boundary points on its communication circle and try to determine its own location.

2.3.1 Assumptions

It is assumed that the mobile node is equipped with a GPS unit or other localization device such that it can find its current location and has enough energy to move and broadcast beacon messages during the localization process.

The sensors nodes are assumed to have rotationally symmetric communication range where each node communicates with nodes that fall within a circle of radius r centered on the node. Although some authors consider that the communication range of a sensor node is not a perfect circle in the physical environment, Beezley (BEEZLEY, 2008) demonstrated that a fair approximation of an isotropic radiator can actually be constructed (CEBIK, 2008).

The perfect circle communication model may not be a realistic description of wireless sensor networks in real word applications, however, it is a valid starting point for modeling purposes and has been used by various researchers such as (SINGH; KHILAR, 2016; HAN et al., 2014; OU; HE, 2013; OU; SSU, 2008; SSU; OU; JIAU, 2005; GALSTYAN et al., 2004)

2.3.2 Localization Principles

Consider the circle in Figure 2 that represents the reception range of a sensor node on the ground. The center of the circle indicates the coordinates of the sensor node while radius r is the largest distance where the sensor node can receive beacon messages from MN. The boundary points A and B are the approximate entry and exit points where the MN passes through the communication circle of the static node. A straight line connecting points A and B can be considered a chord on this circle. There are two possibilities if the value of r and the coordinates of A and B are known. Either $\overline{AB} = 2r$ or $\overline{AB} < 2r$.

In the first case, line $\overline{AB}$ passes exactly over the sensor node (see Figure 2). This situation can be confirmed if the euclidean distance between points A and B equals $2r$. In this case, the center (x_i, y_i) of the circle can be easily calculated using Equation 2.1. It must be noted that the distance between points A and B can be equal to $2r$ (line $\overline{AB}$ passes exactly over the sensor node), irrespective of the MN trajectory.

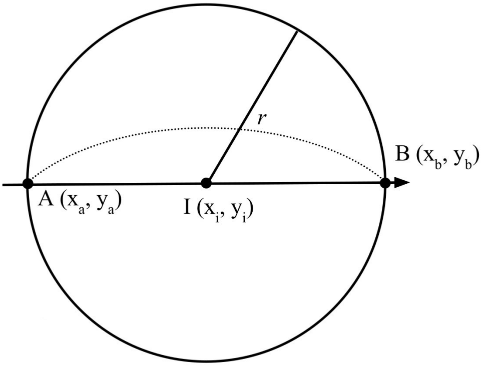


Figure 2 – Localization principle when estimated boundary points lies exactly on the communication circle of the sensor node

In other words, the MN does not need to move exactly on a straight line for this situation to happen. The MN might move on a curved path (see dotted line in Figure 2) and still the chord connecting points A and B might lie exactly over the sensor node.

$$x_i = \frac{x_a + x_b}{2}; y_i = \frac{y_a + y_b}{2} \tag{2.1}$$

In the second case, line $\overline{AB}$ does not pass exactly over the position of the node (see Figure 3) and the euclidean distance between points A and B is less than $2r$. In this situation, in order to find the center of the circle, two new circles with radius r and centers at points A and B respectively, are drawn. These two newly drawn circles intersect each other at two points; one exactly at the location of the node and one above it (see Figure 3). It means that solving the system of linear Equations (2.2 and 2.3) using Cramer's rule or Gaussian elimination method for the values of x_i and y_i, gives two possible locations for the center of communication circle. The sensor node is expected to reside at either one of the estimated locations.

$$(x_i - x_a)^2 + (y_i - y_a)^2 = r^2 \tag{2.2}$$

$$(x_i - x_b)^2 + (y_i - y_b)^2 = r^2 \tag{2.3}$$

$$(x_i - x_c)^2 + (y_i - y_c)^2 = r^2 \tag{2.4}$$

In order to find the correct position out of these two, at least

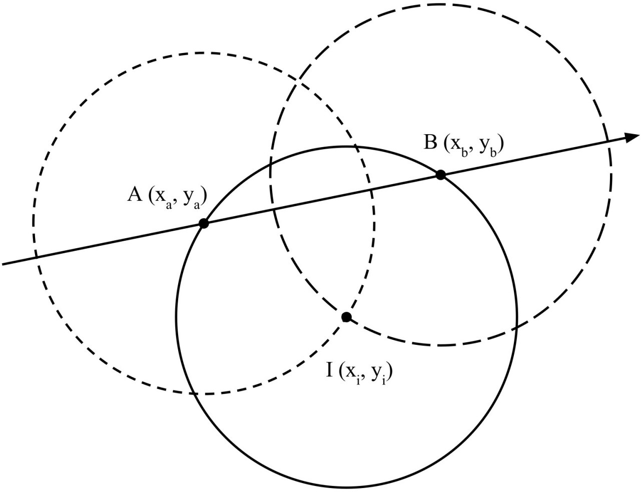


Figure 3 – Localization principle with two boundary points

one other boundary point is needed, with the help of which another circle (see Figure 4) and hence another Equation (2.4) is generated. If points A, B, and C lie exactly on the communication circle of the sensor nodes, then the three circles will intersect each other at a single point (x_i, y_i), which is exactly the required location of the node.

If there is some error in the estimated boundary points (A, B, and C) and they does not lie exactly on the communication circle of the sensor nodes, then the three circles will not intersect on a single point. However, if two out of the three estimated boundary points is error free, the intersection of the three circles results in four possible locations out of which the correct one can be wisely selected. Special technique (for detecting and correcting boundary points error) is therefore needed in order to ensure that at least two of the estimated boundary points is correct so that error in the resultant location of the sensor node is minimized.

To sort out this problem, Equations 2.2 and 2.3 are first solved for x_i and y_i. This results in two possible locations for the sensor node. Equation 2.4 together with Equation 2.2 or 2.3 is then solved in order to find out the intersection points of the third circle (with center at point C) with either the first one (with center at point A) or the second one (with center at point B). If all three boundary points are correctly located on the communication circle of the node, then they

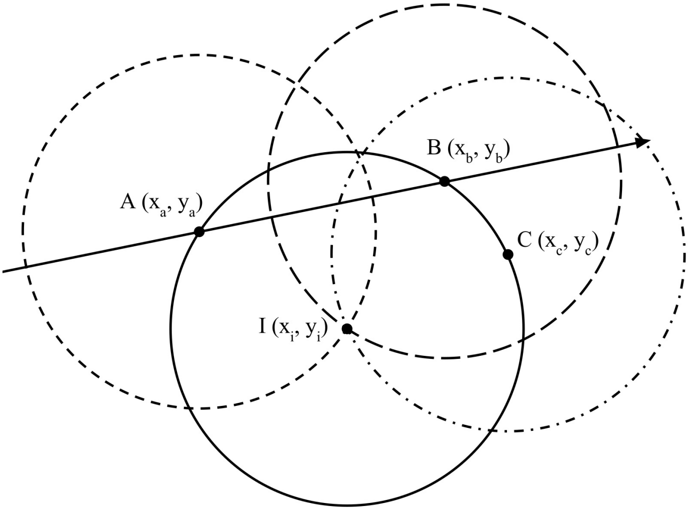


Figure 4 – Localization principle with three boundary points

result in a single intersection point which will be the required location of the node. If there are some errors in the estimated boundary points, then the three circles will not intersect on a single point and will result in four possible locations of the sensor node. Correct location of the sensor node can then be easily identified by verifying that two of these locations are exactly the same or very near to each other and are taken into account. The other two values will be far away locations and will be discarded. In this way, the localization algorithm comes up with a conclusion in estimating the exact location of the sensor node. This mechanism only requires the MN to scan in a proper way the sensing field and to broadcast periodic beacon messages. The ordinary sensor nodes do not spend energy on interaction with their neighbors and do all estimation locally.

It must be noted that in case of an aerial mobile node (UAV), which moves in the sensing field with an altitude of h above the ground level (sensors are assumed to be deployed on the surface of ground), the effective communication radius of static node is smaller than the one calculated on ground. Consider the communication zone of the static sensor node in space as a sphere shown in Fig 5, where the **effective communication radius** (r_e) perceived by an aerial mobile node moving at an altitude h in the upper hemisphere is smaller than the

original communication radius (r_o) perceived by a ground mobile node moving at ground level. If the mobile node is moving at an altitude of h above the ground sensor level, then the value of r in the localizations Equations should be replaced with r_e in order to accommodate the height of the MN. The value of r_e can be easily calculated from Equation 2.5 using Pythagoras theorem. If the value of h is zero (in case of ground mobile node), r_e become equal to r_o. It must be noted here that the altitude for aerial mobile node inside the communication zone of each static node is assumed to be constant.

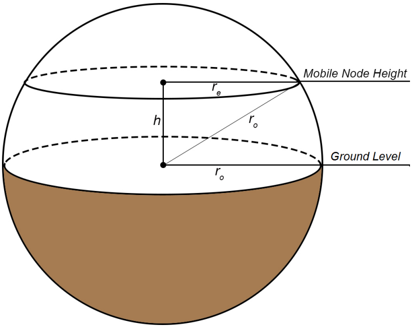


Figure 5 – 3d communication range of a ground sensor node

$$r_e = \sqrt{r_o^2 - h^2} \tag{2.5}$$

Communication radius (r) in the following discussion should be considered as the effective communication radius (r_e) since the value of r_e can be equally used for both aerial and ground mobile nodes.

2.3.2.1 Choosing Boundary Points

In this scheme, at least three boundary points (any entry or exit points) on the communication circle of the sensor node must be obtained in order to establish three circles and hence their intersection points. For this to happen, the MN scans and moves in the sensing area on a predefined trajectory. Equipped with a GPS unit, the MN periodically broadcasts beacon messages that include the current location information and a time stamp. The point where it enters into the communication range and the point it exits the communication range of the sensor node is taken into account. In between entry and exit, there may be periodical beacon broadcasting from MN, but only the entry and exit point are considered as valid boundary points. Each sensor node maintains a beacon table (see Table 1), where all information regarding each received beacon message along with their RSSI values is stored for further use.

Table 1 – Beacon table maintained by each senor node in EGL

Time	**Location**	**RSSI**	**Is Boundary Point**
⋮	⋮	⋮	⋮

The entry point of MN on the communication circle of the sensor node is ascertained with the reception of first beacon message in time t (interval). Here t ensures that it is, in fact, the first entry of MN in the communication circle of the node. Since it might happen that due to some obstacle, a sensor node might not receive beacon messages for a while and after some time when the sensor node start receiving a beacon message again, it might mistakenly assume that it is the entry beacon message. The location information of each beacon message along with its time stamp and RSSI value are stored in the beacon table on each sensor node for as long as it is receiving beacon messages from MN. The instant the sensor node stops receiving beacon messages from MN for a predefined time t, it selects the last received message from the beacon table as being the exit point. Here

$$t \geq \frac{2r}{s}$$

where r is the effective communication radius of the node and s is the speed of mobile node. If a node stops receiving beacon messages from MN, some minimum duration of time t is required for the node to wait and to confirm that the MN has actually gone out of its communication range (and is not hidden due to some obstacles). In order to estimate the correct position of each sensor node, it is necessary to establish at least three boundary points on the communication circle of each node.

2.3.2.2 Mobile Node Scanning

The MN scans the area of interest by performing horizontal or vertical (or both) movements, as illustrated in Figure 6. The scan pattern of the mobile node should be carefully planned and designed to ensure the MN to visit all nodes at least twice in order to find three boundary points (any combination of entry and exit points) on their communication circles. In this way, it is expected that all sensors nodes will be able to find at least three boundary points and hence will be able to estimate their locations. Distance (d_{sl}) in the scanning pattern in Figure 6 does not affect the localization accuracy of the scheme, however, it does affect the percentage of successfully localized node in a particular scanning attempt. Performing only horizontal or vertical scanning require the **distance between scan lines** (d_{sl}) of MN to be equal or less than the communication radius of sensor nodes, as verified in section 2.4.1.6. In this way, all sensor nodes in the scan area can be successfully localized. For distance between scan lines (d_{sl}) greater than r, both horizontal and vertical scanning or multiple scanning attempts might be needed in order to localize 100% of sensor nodes in the scanning area.

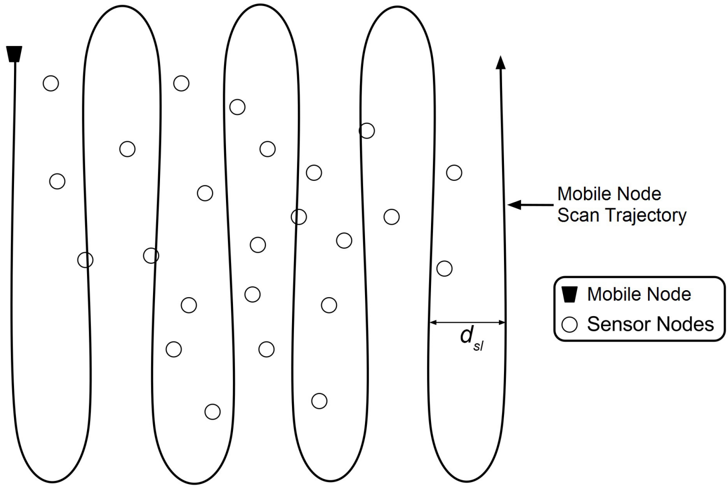


Figure 6 – Mobile Node scanning for EGL

2.3.3 Detecting & Correcting Erroneous Boundary Points

There are various factors influencing the accuracy and performance of the proposed localization scheme but, for sure, the selection of the boundary points is the most important criteria. The estimated location will be more accurate if the selected boundary points exactly lie on the communication circle of the sensor node. However, there may be situations where incorrect boundary points could be chosen, for instance, due to obstacles.

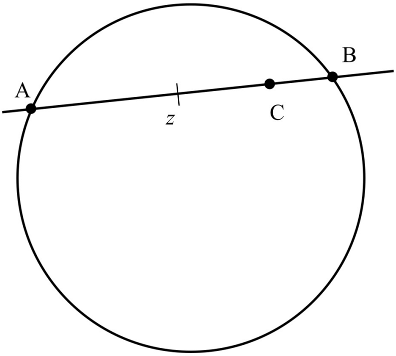


Figure 7 – Incorrect boundary point on the communication circle of a sensor

Obstacles in the sensor field cause radio irregularity and may degrade the performance of localization in the sensor network (JARDOSH et al., 2003; HE et al., 2003b). For instance, in Figure 7 points A and B are the correct boundary points on the communication circle. However, point C may be mistakenly selected as a boundary point if the sensor node does not hear anything from MN after that point. This situation requires an intelligent technique, first to detect errors in the boundary points and then to correct them, if possible.

2.3.3.1 Detecting Erroneous Boundary Points

An enhanced boundary point selection based on the characteristic of concentric circles and RSSI is developed for tolerating the presence of obstacles and inaccurate boundary points. Typically, the received signal strength of a beacon message is inversely proportional to the square of the distance with the sender. If the beacons messages are sent from the same circle (distance), the sensor node will detect that the values of their signal strength are equal. Hence, a sensor node can detect errors in a pair of boundary points from their RSSI values.

By inspecting each pair of boundary points and their RSSI values any anomaly can be detected. It must be noted here that while detecting errors in boundary points a pair of entry and exit point (for every entry boundary point there must be an exit boundary point) should be considered. RSSI is not considered a proper candidate for measuring distance between sensor nodes (HEURTEFEUX; VALOIS, 2012), however, in the present work it is not directly used for distance calculation but for detecting inaccuracies in the pairs of boundary points. The enhanced boundary point selection mechanism can be used for environments without obstacles as well. The scheme improves filtering inaccurate boundary points.

This algorithm requires at least one boundary point (either exit or entry) to be correctly selected on the communication circle of a sensor node. With one correctly chosen boundary point the scheme is able to correct an error in the second boundary point of the same pair as discussed in section 2.3.3.2. With two accurate boundary points, two possible locations can be obtained, out of which one will be correct. The third boundary point and hence the third circle can help in choosing the correct solution among the previously two estimated locations. If the first two boundary points are selected wisely, correct location of the sensor node can still be obtained even if the third point is not accurate.

After two passes of the mobile node, when four boundary points become available to a sensor node, the algorithm first considers their RSSI values. Their signal strength will be the same if they all lie on the communication circle of the sensor node. In this case, any combination of three boundary points with same RSSI values can be used for location estimation. If their signal strength is different (a threshold can be used for stating "*how much different*"), two boundary points that have the same RSSI value and are farthest from the sensor node in terms of distance (with less RSSI), are selected. The farthest boundary point is believed to be the correct one. With two accurate boundary points selected, two intersection points (x_i, y_i and x_j, y_j) are calculated from their corresponding circles, where one intersection point exactly occurs at the location of the sensor node. The correct location between these two can be chosen by using a third boundary point (which may possibly be incorrect). Solving the Equation of the third circle with either the first circle or the second one gives another set of two intersection points (x_k, y_k and x_l, y_l). Now comparing the four intersection points, two will be closest to each other. From the closest pair, the point which was included in the first two solutions is picked. For instance, after getting four intersection points the scheme first find distance between k and i,

k and j, l and i, l and j. Now if k and i are closest, then x_i, y_i is chosen to be the right location of the sensor node since point x_i, y_i belong to the initial two boundary points which were believed to be accurate. In this way, the algorithm came up with a right solution even if the third boundary point was inaccurate.

2.3.3.2 Correcting Erroneous Boundary Points

In the case of an environment with a negligible amount of obstacles, it is highly possible that a sensor node gets at least two accurate boundary points with two MN passes, which are enough for this scheme to find the correct location of nodes. It can be noticed that obstacles inside the communication circle of the sensor nodes do not pose any problem for selecting the accurate boundary points in this scheme. However, there might be cases where all boundary points are inaccurate and have different RSSI values. In this case, the algorithm first chooses a pair of boundary points (an entry and exit point) from the beacon table which contains the farthest boundary point of all and then applies an interpolation technique to rectify the situation.

Considering that a sensor node selects points A and C as boundary points (see Figure 7). This inaccuracy can be easily detected by looking at the signal strength of the beacon message at points A and C. The signal strength of the beacon messages starts increasing from point A till point z and then from point z it again starts decreasing as the MN first gets closer and then again getting far away. Here point C is wrongly chosen by the algorithm because the node did not hear anything from MN after point C due to obstacles. The inaccuracy of point C can be rectified using distance formula and Equation of the chord. First, the distance between point A and the center of the chord (point z) is calculated. The center of the chord can be identified using the RSSI values, the point from where the signal strength started decreasing again. It might happen that the center of the chord is not identifiable due to obstacles; the sensor has no beacon received in the middle of the chord. The center point can still be estimated by taking the mean value of two points on the chord with the same RSSI value. After finding point z, another point (B) is obtained on the same line (chord) such that $\overline{Az} = \overline{zB}$, where B is the new corrected boundary point and will lie on the communication circle of the sensor node. In this case, the unknown boundary point B can be calculated by using distance formula (Equation 2.6) and point-slope formula (Equation 2.7) and solving them for x_b and y_b.

$$Distance\,\overline{AB} = 2(\overline{Az}) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \tag{2.6}$$

$$y_a - y_b = m(x_a - x_b); m = slope\,of\,\overline{Az}\,or\,\overline{AC} \tag{2.7}$$

It must be noted that this interpolation technique works for a pair of boundary points if one of the boundary points (either entry or exit) is inaccurately selected inside the communication range of the sensor node. This location scheme assumes that at least one of the boundary point out of the four boundary points is accurately selected in order for the detection and correction technique to work. Whenever a sensor node detects different signal strength values for entry and exit points, the point with the least signal strength is assumed to be the correct one and the point with higher signal strength is considered to be erroneous and is tried for rectification. It must also be noted that the path of the mobile node is assumed to be fairly straight inside the communication range of a sensor node in order for the interpolation technique to correct an error in a boundary point for that node.

2.4 PERFORMANCE EVALUATION

The performance evaluation of the proposed EGL scheme is done by means of simulations. Such simulations are conducted in the Omnet++ 4.41 simulator with Inetmanet 2.0 framework. The simulated scenario consists of a 400 m by 400 m sensing field, with 25 sensor nodes randomly deployed. Each sensor node is forced to stay active during the localization process. After estimating its location, the sensor node enters a low power mode. The transmission technology used for all nodes including the mobile node is IEEE 802.15.4 and the radio propagation model was based on the free space model.

When the localization mission begins, the mobile node starts scanning the sensing area on a pre-defined path as discussed in section 2.3.2.2. The mobile node begins broadcasting beacon messages at some distance (outside the range of all nodes) from the sensing area. Otherwise, the sensor node will mistakenly take the first message as a boundary point. The mobile node traverses the sensing field using the scan line pattern shown in Figure 6.

Average location error (E_r), which is the average distance between the estimated location (x_{e_i}, y_{e_i}) and the actual location (x_i, y_i) of all sensor nodes, is used to evaluate the performance of the localization mechanism.

$$E_r = \frac{\sum \sqrt{(x_{e_i} - x_i)^2 + (y_{e_i} - y_i)^2}}{\#\, of\; sensor\, nodes} \tag{2.8}$$

For simulating environments with obstacles, a probabilistic technique is introduced, allowing to emulate the occurrence of interferences, as follows. Along the path of the mobile node, for certain distances (and hence time), the mobile node does not broadcast any beacon messages. During this time, the sensor nodes will hear nothing as if there are some obstacles in between the MN and sensor, completely blocking the signals. This is quite pessimistic since in practice an obstacle only blocks signals to some particular node, but in this case, all nearby nodes will hear nothing as if there are obstacles in front of all the surrounding nodes. The entire trajectory of the MN scan pattern is divided into 50 m long slices. An obstacle, with randomly chosen size from 2 m to 10 m is then placed on each slice at some random position. This mimics an environment where a 2 m to 10 m obstacle is placed in front of each sensor node after every 50 m distance along the path of MN. It must be noted that the interpolation technique was used only for the environment with obstacles and that it was not applied in the case of the obstacles-free environment during simulation experiments.

2.4.1 Parameters Studied

The adopted parameter settings with their default values (in parentheses) during simulations are summarized in Table 2. The following subsections detail how such parameters were varied and studied during the conducted performance analysis.

2.4.1.1 Beacon Interval

As a performance measure, the intervals between beacon messages from MN have a considerable effect. More beacon messages and less beacon broadcasting interval leads to efficient and accurate selection of entry and exit points on the communication circle of each node.

Table 2 – Parameters setting for simulation

Parameters	**Values**
Beacon Interval	0.01, 0.03, 0.05, 0.07, (0.1) 0.3, 0.5, 0.7, 0.9, 1 s
Mobile Node Speed	1, 2, 3, 4, (5), 10, 15, 20 25, 30 m/s
Communication Range	(50), 60, 70, 80 m

Figure 8 compares average location error for a periodical broadcasting schemes with varying beacon intervals. Reducing the beacon interval improves localization accuracy of this schemes. Using 0.1 s beacon interval, the error was less than 1 m. The error was further reduced to 0.24 m when the beacon interval was further reduced to 0.02 s (about 10 beacons per sec). Communication overhead in EGL scheme only includes broadcasting beacon messages by mobile node. It must be noted that the beacon overhead grows as the beacon interval is decreased. The algorithm also performs well in the presence of obstacles where the lowest localization error is 0.34 m when the beacon interval is 0.01 s.

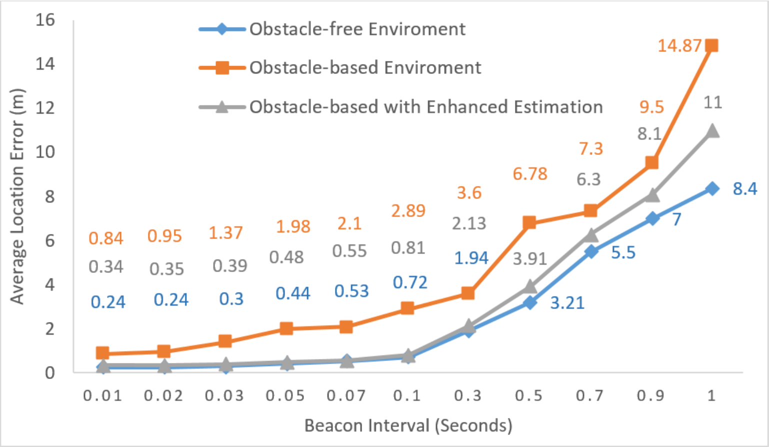


Figure 8 – Performance with varying Beacon interval at the speed of 5 m/s

2.4.1.2 Mobile Node Speed

Increasing the MN speed increases the localization error if the beacon interval is kept constant (see Figure 9). This is due to the fact that with higher speeds, fewer beacons are sent in a fixed time which results in inaccurate boundary points. To maintain localization accuracy, beacon intervals must be decreased as the speed of the MN is increased.

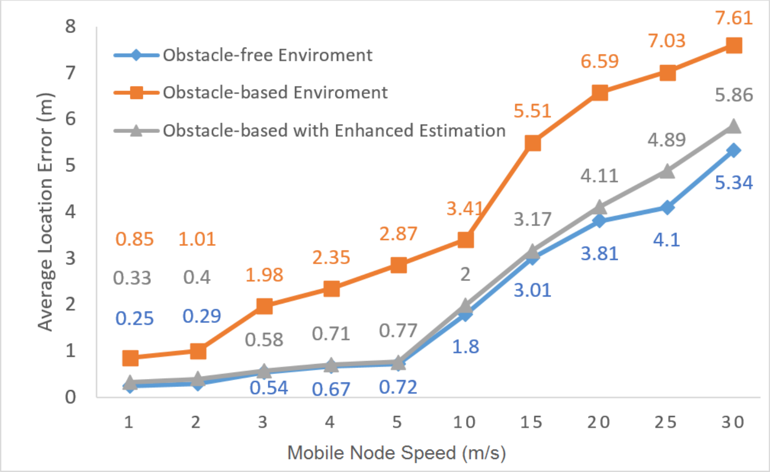


Figure 9 – Performance with varying MN speed for a beacon interval of 0.1 s

2.4.1.3 Beacon per Meter

After analyzing the performance of localization error with different speed and beacon interval, it is apparent that it is actually the number of beacon per meter that is affecting the average localization error. To maintain the original localization accuracy with faster MN, it is necessary to lower the beacon interval correspondingly. The objective is to broadcast fixed number of beacon messages in a given distance (e.g., 10 beacons/m). From Figure 8 it can be noticed that 10 beacons per meter result in a minimum localization error of 0.24 m in the obstacle-free environment and 0.34 m in the obstacle-based environment. Maintaining the same number of beacons per meter for different mobile node speeds results in localization errors as shown in Figure 10.

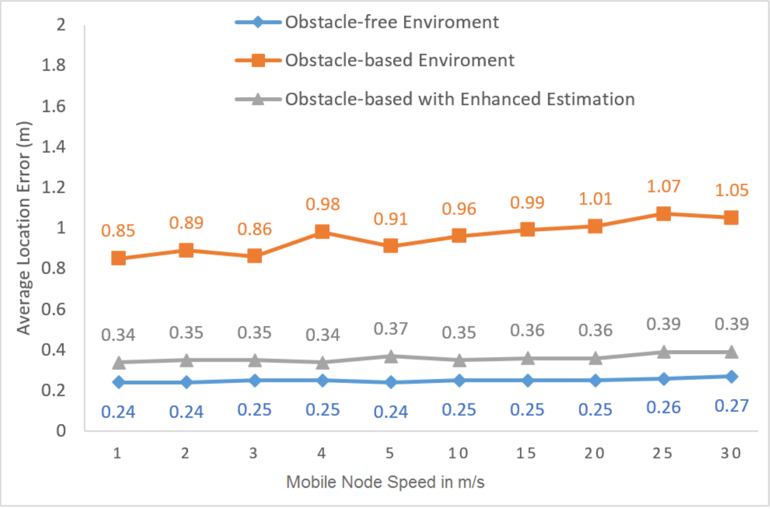


Figure 10 – Performance with varying MN speed for 10 beacons per meter

The relation between optimum beacon interval and desired MN speed is presented in Figure 11.

2.4.1.4 Sensors Communication Range

Communication range has very little effect on the estimation of localization. Figure 12 shows the performance of four different radio ranges where location errors are slightly decreased with the larger transmission range.

Localization error of EGL algorithm roughly varies from 1% to 15% of communication ranges depending on beacon interval, MN speed, and nature of obstacles.

2.4.1.5 Mobile Node Trajectory

In an obstacle-free environment, the impact of path or trajectory of the MN on the localization error is none. Regardless of the MN

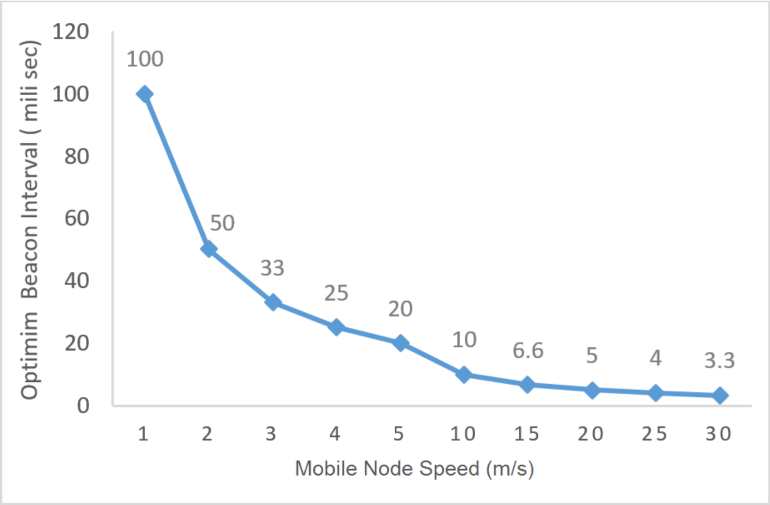
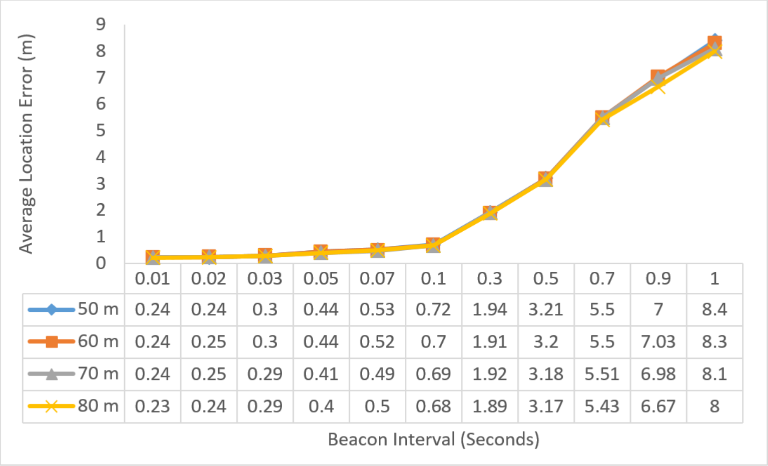


Figure 11 – Mobile Node Speed vs Optimal Beacon Interval



	0.01	0.02	0.03	0.05	0.07	0.1	0.3	0.5	0.7	0.9	1
50 m	0.24	0.24	0.3	0.44	0.53	0.72	1.94	3.21	5.5	7	8.4
60 m	0.24	0.25	0.3	0.44	0.52	0.7	1.91	3.2	5.5	7.03	8.3
70 m	0.24	0.25	0.29	0.41	0.49	0.69	1.92	3.18	5.51	6.98	8.1
80 m	0.23	0.24	0.29	0.4	0.5	0.68	1.89	3.17	5.43	6.67	8

Figure 12 – Performance with varying radio Range in obstacle-free environment (MN speed 5 m/s)

trajectory, whether it is a straight line or curved line (due to the wind etc. in the case of an aerial mobile node), as shown in Figure 13, the algorithm is always able to find accurate beacon points for each sensor

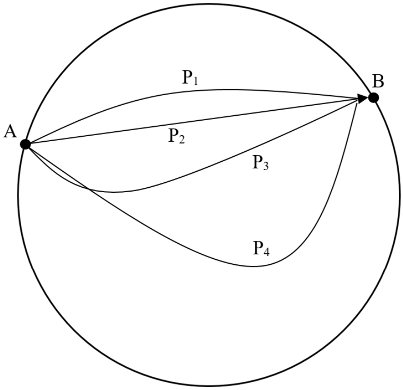


Figure 13 – Possible MN Trajectories

node. However, in the case of environments with obstacles, the MN trajectory should be fairly straight (no greater than 60 degree turns) in order for the interpolation method to work efficiently and rectify an incorrect boundary point, if any. For instance, the interpolation scheme described before works well with trajectories P_1 and P_2 in Figure 13 but may not interpolate accurately trajectories P_3 and P_4. During simulation, the MN moved with a straight line while crossing the area of interest, as shown in Figure 6.

2.4.1.6 Mobile Node Scanning Pattern

Varying the distance d_{sl} between the scan lines of MN trajectory has no effect on the average localization error. However, it does affect the number of sensor nodes that are successfully localized after completing one vertical scan as shown in Figure 6. Figure 14 shows the percentage of sensor nodes that are able to get at least three boundary points on their communication circle after completion of one vertical scan. In the case of an environment with obstacles, some portion of the nodes is not able to find accurately at least three boundary points. However, after using the enhanced estimation mechanism including the interpolation method, this mechanism is able to increase the number of successfully localized nodes. In an obstacle-free environment, a safe value for separation d_{sl} between MN scan lines is r, while in the case of obstacles, the separation should be less than r depending on the nature and number of obstacles. Similarly, a vertical followed by a horizontal scan pattern can also be used for efficient localization. The motive should be to ensure that the MN passes through each sensor at least two times in order to get enough boundary points required for a correct localization estimation.

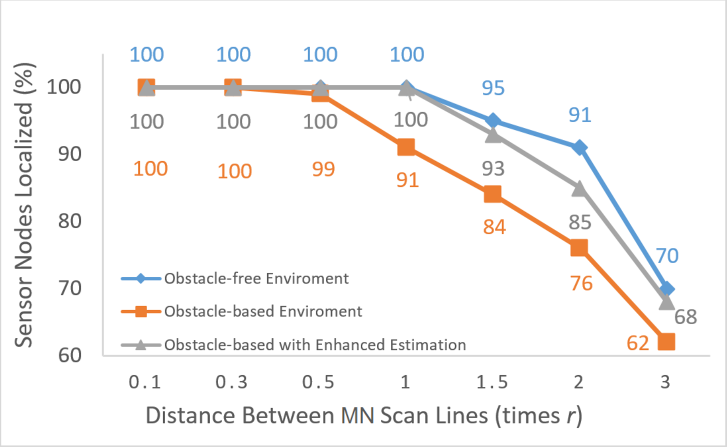


Figure 14 – Performance with varying distance between MN scan pattern lines

2.4.2 Preliminary Practical Experiment

In order to validate the localization scheme and test its applicability, some preliminary practical experiments were performed. Since the basic principle and applicability of our approach does not depend on the underlying communication model of the unknown devices, smartphones were used in the initial practical experiments, mainly due to simplicity and ease of use.

Two smartphones were used as the unknown nodes and one was used as a mobile node. First, two locations were randomly chosen about 50 m apart in an open sports field in the premises of UFSC. One Huawei P8 Lite mobile phone was placed at each chosen location. The location of each static phone was calculated with the help of the built-in GPS and Google Maps installed on each smartphone. Afterwards, a rough map and nearly rectangular path were marked and used as the scanning pattern path for the mobile node (see Figure 15). One person

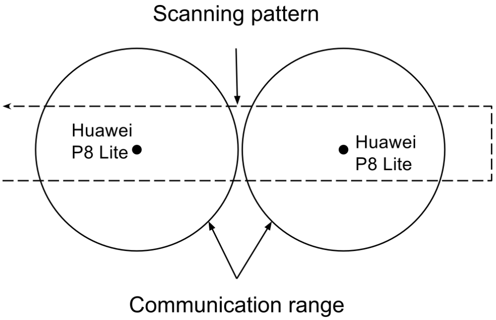


Figure 15 – Preliminary Practical Experiment for EGL

with another smartphone of the same model moved slowly along the scanning path, while continuously transmitting its current coordinates via Bluetooth to the static nodes. The transmission and receptions of the mobile node coordinates were performed manually[1] using the built-in Bluetooth technology in the smartphones. All beacon messages were logged and the first and last messages received at the static nodes were considered as boundary points on its communication circle. An online mapping tool[2] was used to find the required intersection points using the boundary points. Similarly, a MATLAB function[3] was used to find the distance (localization error) between the estimated location and the one found with the help of GPS and Google maps at the initial stage.

One complete round of the mobile node or establishing of at least three boundary points for each static node was considered one experimental run. Three experimental runs were performed (with the unknown nodes at the same positions, but slightly changing the path of the mobile node).

The boundary points, the positions of unknown nodes discovered through GPS and Google maps (called GPS position), the estimated positions, and the localization errors are tabulated in Table 3. It can be seen that the average location error in this small practical experiment turns out to be 0.91m. Such localization accuracy can be acceptable in most outdoor applications, establishing the viability and applicability of the proposed approach in practical scenarios.

---

[1] An application which automatically transmits location coordinates was neither found on the Google Play Store nor developed during the limited available time for performing the preliminary practical experiments

[2] Available at https://www.mapdevelopers.com/draw-circle-tool.php

[3] distInMeters = distance(lat1, lon1, lat2, lon2, earthRadiusInMeters)

2.4.3 Analysis of Localization Accuracy

The *boundary point error* (e) is defined as the distance between the selected boundary point (Á) and the nearest point on the communication circle (A), as shown in Figure 16. The maximum theoretical boundary point error e is the product of the beacon interval and the speed of the mobile node. For instance, if the mobile node is moving with 1m/s and the beacon interval is 0.1s, the worst boundary point error could be 0.1m. Boundary point errors, if exist, propagate down the calculation process and generate error E_r in the final estimated point of the unknown node.

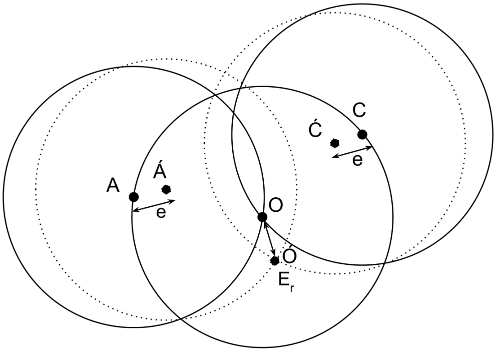


Figure 16 – Maximum theoretical location error in EGL

Using MATLAB and exhausting all the possible combinations of e in the x and y coordinates of boundary points, the maximum error in the final estimated location can be found. For instance, if the boundary points $(x1, y1)$ and $(x2, y2)$ result in the intersection point of (X, Y), then $[x1, y1, x2, y2] + e \geq 0$ results in $[X, Y] + e \geq 0$. Similarly, $[x1, y1, x2, y2] - e$ results in $[X, Y] - e$ and $[x1 + e, y1, x2 + e, y2]$ result in $[X + e, Y]$. The maximum possible deviation of the estimated final location is $[X \pm e, Y \pm e]$. So the *maximum theoretical location error* could be the distance between error-free coordinates and error based coordinates. which is

$$Max\,Er = \sqrt{(X - X \pm e)^2 + (Y - Y \pm e)^2} = e\sqrt{2}$$

Table 3 – EGL Practical Experiment Results

Parameters	**Node 1**	**Node 2**
Experimental Run 1		
GPS Location	-27.605308, -48.519078	-27.605505, -48.519566
Estimated Location	-27.605300587816725, -48.519083059281571	-27.605507969375951, -48.519558480829461
Beacon point 1	-27.605114, -48.518879	-27.605307, -48.519372
Beacon point 2	-27.605307, -48.519377	-27.605578, -48.519858
Beacon point 3	-27.605531, -48.519261	-27.605537, -48.519266
Location Error	0.96m	0.81m
Experimental Run 2		
GPS Location	-27.605308, -48.519078	-27.605505, -48.519566
Estimated Location	-27.605299399382595, -48.519093788117630	-27.605513360074962, -48.519562536683736
Beacon point 1	-27.605121, -48.518862	-27.605352, -48.519319
Beacon point 2	-27.605356, -48.519382	-27.605594, -48.519835
Beacon point 3	-27.605487, -48.519299	-27.605466, -48.519273
Location Error	1.82m	0.99
Experimental Run 3		
GPS Location	-27.605308, -48.519078	-27.605505, -48.519566
Estimated Location	-27.605303007440735, -48.519075715747650	-27.605503852619975, -48.519567901101880
Beacon point 1	-27.605288, -48.518773	-27.605272, -48.519416
Beacon point 2	-27.605220, -48.519366	-27.605393, -48.519842
Beacon point 3	-27.605559, -48.519211	-27.605597, -48.519283
Location Error	0.59m	0.23m

2.4.4 Comparison with other Approaches

A comparison of our approach in terms of performance and features with few other popular mobile beacon-based solution is given in Table 4. Estimated execution time represents the total estimated time required by all sensor nodes to compute their locations. Simulation execution time represents the total time spent by all sensor nodes to compute their locations during the reported simulation experiment. In Table 4, *MAP* refers to the work in (SSU; OU; JIAU, 2005), *Centroid* refer to (BULUSU; HEIDEMANN; ESTRIN, 2000), *Constraint* refer to (GALSTYAN et al., 2004), *Munir* refer to (MUNIR; BIN; JIAN, 2007), *Sichitiu* refer to (SICHITIU; RAMADURAI, 2004), *Singh* refer to (SINGH; KHILAR, 2016), and *EGL* refer to the proposed localization approach in this thesis. It must be noted that the *Centroid* and *Constraint* approaches were modified to perform localization with mobile beacons by (SSU; OU; JIAU, 2005).

The proposed EGL solution is scalable and distributed since each sensor node only requires to listen to the beacon messages from the mobile node without any interaction with their neighbors and all computations are done locally on each node. The approach scales well to any number of nodes and any size of the network, provided that the scanning pattern of the mobile node is properly adjusted to cover the entire sensing area. In the case of centralized computations, the data needs to be gathered from the entire network which would likely consume significant resources in very large networks.

The location estimates are competitive in comparison to other approaches that typically result in accurate localization but usually require additional hardware (HIGHTOWER; BORRIELLO, 2001). For instance, the approach used in (OU, 2011) require four directional antennas, (PATWARI et al., 2005) needs to find the angle of arrival using an array of antennas, while (GUERRERO; ALVAREZ; RIVERO, 2010) uses an aerial mobile node equipped with a rotary and tilting directional antenna along with the GPS unit.

The computational cost of our approach was not directly measured. However, computing coordinates on each sensor node are entirely possible even on devices similar to the Berkeley motes (HILL et al., 2000). Software optimizations together with a microcontroller (even the one used in the first generation of MICA Berkeley motes at 16MHz) can reduce the computation time to process the operations of multiplications and divisions (SICHITIU; RAMADURAI, 2004). In short, considering the fact that the localization process is only done once (immediately after deployment), spending a couple of minutes on localization is perhaps reasonable.

Algorithm 1: EGL: Efficient Geometry-based Localization

```
Mobile Node: Periodically generate and broadcast beacon message $B_i$
Sensors    : Listen for beacon message $B_i$
if a sensor node first receives $B_i$ in $t$ time then
    (Re)Start timer $T$, which runs till $t$ time;
    if $T$ is running then
        $B_i$ is not a boundary beacon; Save in beacon table;
    else if $T$ is not running then
        $B_i$ is a boundary beacon; Save as boundary (entry) point;
    end
end
if timer $T$ expires then
    Mark last $B_i$ from beacon table as boundary (exit) point;
    if Distance between entry and exit points equals $2r$ then
        Calculates final location using Equation 2.1 ;
        GoTo power saver mode
    end
end
if Four boundary points are available then
    if RSSI of at least two farthest points are equal then
        Uses those points to generate two circles;
        Uses geometric Equations to calculate their two intersection points;
        Uses help from third boundary point (the closest in term of RSSI) to choose the correct location;
        GoTo power saver mode;
    else if RSSI of all boundary points are different then
        Picks a pair of entry and exit points (with a farthest point) and use interpolation;
        Uses those points to generate two circles;
        Uses geometric Equations to calculate their two intersection points;
        Uses help from third boundary point to choose the correct location;
        GoTo power saver mode;
    end
end
```

Table 4: EGL vs other mobile beacon-based approaches)

Scheme / Feature	**MAP**	**Centroid**	**Constraint**	**Munir**	**Sichitiu**	**Singh**	**EGL**
# of Mobile Nodes	≥1 [4]	≥1 [4]	≥1 [4]	≥1 [4]	≥1	≥1	1 [5]
Beacon Interval	Depends[6] 1/m [7]	Depends[6] 1/m [7]	Depends[6] 1/m [7]	Depends[6] 10/m [7]	NA	Depends[6] 0.2/m [7]	Depends[6] 10/m [7]
Beacon Collision	Yes	Yes	Yes	Yes	No	No	No
Estimated Execution Time	Variable[8]	Variable[8]	Variable[8]	Variable[8]	Variable/Fixed	Variable[8]	Fixed [9]
Simulation Execution Time	5697s [10]	13362s[10]	13362s[10]	NA	NA	NA	106s[11]
E_r vs Radio Range	Decrease[12] (2-8%)[13]	Increase[10] 30%[13]	Increase[10] 20%[13]	Decrease[10] 4-10%[13]	NA	Increase[10] ≥10%[13]	Decrease[10] 0.5-10%[13]
Minimum E_r	0.68m [14]	3.01m	2.12m	2m	2m	2m	0.24m [15]
Correct Beacons[16]	3	200[17]	200 [13]	NA	3-9 [18]	2	3

---

[4] Localization accuracy depends on the number of mobile nodes. Increasing the number of mobile nodes also increase beacon overhead

[5] Localization accuracy in both obstacle free and obstacle-based environment does not depend of the number of mobile nodes

[6] Beacon interval (beacon/s or beacon/m) affect the localization accuracy

[7] Value used for lowest localization error during simulation experiments

[8] Random, since mobile nodes move in random fashion. Depends on the network density, size and shape of the network area

[9] Deterministic, depends on the length of the scanning pattern and speed of mobile node. Don't depend on the network density

[10] MAP, Centroid and Constraints took 5697s, 13362s and 13362s respectively for 319 nodes in 100m2 area with 6 mobile nodes

[11] 1 mobile node took 640s for 25 nodes in a 400m2 area. Since, the execution time depends only on the path of the scanning pattern, it is speculated that for 319 nodes in a 100m2 area with 6 mobile nodes, it will take 640/6=106s

[12] E_r decrease/increase with larger radio ranges

[13] Percentage of localization error with respect to radio range

[14] E_r depends on the threshold for selection of the chord's length, radio range, beacon interval, and speed of mobile nodes.

[15] E_r depends on the length of scanning pattern, beacon interval, and speed of mobile node.

[16] Number of appropriate beacon messages needed for each sensor node for localization

[17] Each sensor had to obtain 200 beacon messages for localization according to the implementation in (SSU; OU; JIAU, 2005)

[18] At least 3 beacon messages are required for the scheme to work but on the average, 9 will be required for better results (2m)

2.5 CHAPTER CONCLUSIONS

This chapter described a geometry-based localization scheme for static sensor nodes in WSN. First, a Mobile Node is used to find boundary points on the communication circle of the sensor nodes spread within an area of interest. Based on the estimated coordinates of the boundary points, together with elementary geometry and algebra, each sensor node calculates its position without additional interactions with their surroundings. The proposed algorithm is scalable, accurate, distributed, and power efficient since the Mobile Node only broadcasts beacon messages and all computation is done locally on each sensor node. Several enhancements, including adjusting mobile node beacon scheduling, speed selection, detection and correction of boundary points errors were also explained for tolerating signal blockage and irregularity due to obstacles. The algorithm was successfully implemented and evaluated using simulation in Omnet++ tool. The preliminary practical results also show fine-grained accuracy and are competitive to other approaches that typically require additional hardware on each sensor node.

3 ADAPTIVE SPEED CONTROL

Habitat monitoring was one of the earliest applications of sensor networks. In Great Duck Island (SZEWCZYK et al., 2004), researchers monitored the behavior of petrels, especially about how they use burrows both in short-term and long-term periods. They also monitored the environmental parameters inside and outside of burrows. Another example is the ZebraNet project (JUANG et al., 2002), in which the behavior of zebras including long-range migration and inter-species interactions is monitored using tracking collars.

Environmental monitoring is another area that sensor networks have been widely used. For instance, meteorological and hydrologic processes at high altitudes (LUNDQUIST; CAYAN; DETTINGER, 2003), long-term glacial movement (MARTINEZ; HART; ONG, 2004), temperature and humidity in the forest (BATALIN et al., 2004), and volcanic activities (WERNER-ALLEN et al., 2006).

Besides, there are target tracking applications, where the objective is to localize a target by different techniques using multiple sensors capable of measuring distance or angle of the target (e.g., (CHU; HAUSSECKER; ZHAO, 2002; SIMON et al., 2004). Target tracking applications are qualitatively different from habitat and environmental monitoring applications in the sense that it essentially requires collaborative information processing among sensors (SUGIHARA; GUPTA, 2008). However, in all of these applications, the high-level objective is to bring the sensor data from a distributed field of sensors to a base station.

3.1 DATA COLLECTION IN WSNS

Irrespective of the nature and type of application, data collection remains as one of the basic and main function of WSNs (WANG et al., 2015). Data collection methods can be broadly classified into two categories. In the first case, the sink node is fixed and remains static throughout data collection process. Sensor node transmits its measured data to the sink node either through one-hop or multi-hop routing. Single-hop transmission is not always viable since it is not possible for every node to send its data directly to the base station and even if it is possible for a node, it is not an energy efficient option (HEIDEMANN; YE, 2005). Multi-hop routing is a good option for nodes which cannot directly transmit to the base station due to limited communication range or energy. However, for multi-hop transmission to occur, the network needs to be connected since multi-hop routing requires relay nodes or other sensor nodes for messages forwarding. Multi-hop forwarding is more energy-efficient than long range communications, however, it cannot be used when the network is disconnected. In addition, for sparse networks, even if the network is not disconnected, energy consumption is high because the distance of each hop tends to be long. A possible workaround for these cases is to deploy additional nodes to maintain the connectivity and reduce each hop distance, but this may not always be possible.

Another issue with multi-hop routing is that the energy consumption throughout the network is not uniform and nodes near the base station deplete their energy much faster since they also forward a large amount of data received from other nodes. Moreover, the nodes need to be capable enough to handle multi-hop communication pattern. This becomes a problem for some applications such as (TODD et al., 2007), which uses RFID-based sensor nodes. The energy issue in WSN is very important since they are usually deployed in remote areas with limited battery power and they need to last for a longer time. This type of data collection is not always the best, particularly in large and sparse WSN, where the network is not always connected.

The second and most popular choice nowadays is to use a mobile data collector (MDC) for data collection, particularly in large and sparse WSN. Such MDC could be a ground robot or an aerial vehicle which moves to different areas of the network and collects data directly from sensor nodes thereby spreading the energy consumption more uniformly (WANG et al., 2005). In addition, disconnected networks or sparse networks are not an issue for MDC approaches, since the communications among nodes are not necessary. This type of data collection is very useful in sparse WSN where sensors are not always connected to each other. Aerial MDC has an additional advantage of moving flexibility, as it can go deep in scenarios such as swamps, flooded areas, and (above) thick forests etc.

3.1.1 Data Collection with MDC

Suppose, in the network, there are one or more mobile nodes that are capable of traversing the network area, communicating with sensor nodes, and carrying data. In this case, the best idea is to use these mobile nodes as routers from sensor nodes to the base station. These mobile nodes can either consume the collected data itself or pass it to a base station. In the later case, the mobile node (MDC) starts from the base station, moves to different areas in the sensor field while collecting data from sensor nodes, and eventually comes back to the base station to flush the collected data.

3.1.2 Optimizing MDC Speed

A drawback of using an MDC is that it increases data delivery latency the time taken from data measurement at a sensor until it is delivered to the base station. This data delivery latency is dependent on the physical motion of MDC, which is relatively slow compared to the speed of data transmitted in multi-hop forwarding.

Similarly, the total amount of data collected is an important parameter in networks where sensors have some latency or buffer constraints applied. In this case, the MDC must collect the maximum possible data within the given latency constraints or before buffers overflow and hence should remain in contact with each node for the right amount of time. In scenarios where static nodes have variable sampling rates (some are generating at a slow rate while other at a faster rate), or where the network is large and sparsely deployed, the data collection requirement in not uniform.

For this reason, optimizing the MDC speed becomes an important matter. In all these situations, conditions for data collection can be made more favorable and the performance of the network can be increased further by properly adjusting the motion of the MDC. By adjusting the speed of the MDC, the actual contact time with each static node is changed, which if best tuned, results not only in higher data collection rate but also lower MDC traversal time. Minimizing the MDC traversal time does not only reduces the expenses related to MDC mission but also minimizes data transfer delays, buffer overflows, and increases the network lifetime.

The remaining parts of this chapter are organized as follows. Section 3.2 presents an overview of the related works with an outline of the open issues that need to be addressed. Section 3.3 details the proposed solution for using a mobile node as an MDC. Simulation results are presented and discussed in Section 3.4, which highlights the advantages and benefits obtained by using the proposed solution. Conclusions and hints to future work are presented in Section 3.5.

3.2 RELATED WORKS

Mainly three types of mobility, namely random, predictable, and controlled, are exploited in WSNs for improving performance. In most cases, mobile components have random mobility with partially predictable patterns or even deterministic but not fully controlled. For instance, random mobility is considered in (GROSSGLAUSER; TSE, 2002) for improving data capacity in WSNs. Random motion of mobile nodes was also explored in (SMALL; HAAS, 2003), where whales were used to carry data in the network. However, the latency of data transfer cannot be bounded in such cases. Similarly, predictable mobility was considered in (CHAKRABARTI; SABHARWAL; AAZHANG, 2003b), where mobile nodes (base station) are assumed to be onboard of public transportation shuttles that visit sensor nodes along its path for collecting data.

Random or predictable mobility can be advantageous in WSN but full potentials of mobility can be realized only in the presence of controlled mobile elements. Controlled mobile nodes can actively change their location because they have the ability to dynamically control their trajectory and speed (FRANCESCO; DAS; ANASTASI, 2011). In the case of controlled mobility, some of the operation of WSNs can be significantly enhanced and simplified (FRANCESCO; DAS; ANASTASI, 2011). For instance, the discovery process of mobile nodes can be improved and simplified, by controlling the mobile nodes so that they visit the static nodes at specific times while at the same time move slower/faster or even stop at nodes.

In this regard, several schemes have been proposed, such as in (KANSAL et al., 2004b) and (SOMASUNDARA et al., 2006), where approaches targeted for controlling mobility are defined. For instance, in (KANSAL et al., 2004b) the authors propose a solution for controlling the speed of a ground mobile node in WSN. In the first case, when the mobile node enters the communication range of a static node that has some data to send, it stops there until the collection of all buffered data. In the second case, the speed of mobile node is changed according to the number of encountered nodes and the percentage of collected data. Different groups of nodes are made according to the amount of data collected, such as low, medium, or high. The mobile node moves slowly in the group with a low percentage of collected data while it moves faster when it is in communication range with the nodes with a high percentage of collected data.

The data collection scheme can be tied to the WSN architecture deployment, as one can see in (KONSTANTOPOULOS et al., 2015). In such work, the WSN is deployed in an overlapping manner, in which all nodes can be reached. The WSN deployment is important because it provides a solution for a better data gathering scheme. Its main idea is to select a set of rendezvous nodes which will be visited by mobile sinks. The rendezvous nodes act as caching nodes that temporarily store the data from other nodes. The other nodes select the best path to reach a rendezvous node. The metric to select the best path to reach the rendezvous node is the node remaining energy. The work suggests that the use of multiples mobile sinks and an appropriate set of rendezvous nodes could decrease the data delivery delay while prolonging the network lifetime.

Some studies using MDC (KANSAL et al., 2004a; XING et al., 2007) have simplified the problem by using simple models for mobility and communications. Other examples tackle the problem as a scheduling problem (e.g., (SOMASUNDARA; RAMAMOORTHY; SRIVASTAVA, 2007)) while some assume constant MDC speed (e.g., (MA; YANG, 2007)).

After analyzing the related works, several issues/limitations are observed. For instance, there are no specific solutions for controlling and optimizing the MDC speed in order to maximize the data collection rate and to minimize delays in WSNs. The technique proposed in (KANSAL et al., 2004b) is particularly adapted for ground robots with only 3 supported speed variations (stopped, 0.5m/s, and 1m/s). Mobile robots must be able to move at any speed in a given acceptable range of speeds so that they could fully obtain the advantages provided by speed optimization policies, of which the most important advantage is to adjust the contact time with the static node.

This chapter proposes a method to exploit controlled mobility for promoting an intelligent data collection scheme targeting sparse WSNs. It presents an adaptive algorithm along with a discussion of control parameters that the MDC uses for continuously and autonomously optimizing its motion. The proposed solution can be adapted to all types of MDCs, including aerial and ground robots, with varying speeds (simulations results shown for up to 30m/s) and radio models, such as 802.15.4 and 802.11.

This work focuses on autonomously optimizing the motion of an MDC with controlled mobility. It must be noted that it does not consider optimizing the MDC trajectory since this is a related but in itself a different issue to be dealt with. The term MDC is solely referred to a controllable mobile node used for data collection which takes a trajectory as an input and then starts data collection by traveling over a path with an adaptive and optimized speed.

3.3 ADAPTIVE SPEED CONTROL

3.3.1 System Model

Considering a wireless sensor network where several static nodes are deployed in a remote area for monitoring an environment of interest. The static sensor nodes are organized into non-overlapping clusters on the basis of the distance from each other. The sensor nodes sense their environment and save data in their buffers. Each sensor transmits data to its corresponding CH, which in turn saves the data and eventually sends them to the MDC whenever it is in communication range. The cluster heads are responsible for coordinating the transmissions of the cluster members. Cluster members send data to their corresponding CH using single hop transmission while CH sends the collected data to the MDC using a single hop, thus minimizing collisions and message losses. In this case, the MDC should visit each CH individually and come within the closest possible distance to it.

In WSNs, the mobility pattern (trajectory and speed) followed by MDC has a significant impact on the data collection process (FRANCESCO; DAS; ANASTASI, 2011). Conditions for data collection can be made more favorable and the performance of the network can be increased further by properly adjusting the motion of the MDC. The aim of this work is to develop an intelligent and adaptive motion control for MDC in order to autonomously adjust its speed during the data collection process.

The total amount of data collected is an important parameter in networks where the static nodes have variable sampling rates (some are generating at a slow rate while other at a faster rate) or if sensors have some latency constraints applied. In this case, the MDC must collect the maximum possible data from the static nodes within the given latency constraint and hence should remain in contact with each CH for a proper amount of time. In addition to that, the static sensor nodes may have a limitation of finite buffer and the MDC needs to collect the data before buffers overflow might occur. This means that certain data has to be collected within the given time constraint which may require the MDC to best use its traversal time. The term *traversal time* is defined to be the time that the MDC requires for completing one round by visiting all CHs in the sensing area, irrespective whether it is collecting data or not. To tackle the situation, the speed of the MDC should better be dynamically controlled. It is suggested for the MDC to move slower or even stop (if it can and if it has to) in areas where more time is needed to collect the required data and to speed up where there are no such constraints. By adjusting the speed of the MDC, the sojourn time (time spent in actual data transfer with each CH) is changed, which if best tuned, results in higher data collection rate and lower MDC traversal time. Minimizing the MDC traversal time does not only reduces the expenses related to MDC mission but also minimizes data transfer delays. An adaptive speed control is presented in this chapter for the MDC to meet its time constraints requirements. The MDC calculates its optimal speed on the fly using parameters from the environment. Adaptive speed control is also associated with reduced buffer overflows and increased network lifetime, as shown for ground robots in (KANSAL et al., 2004b).

3.3.2 Premises for Adaptive Speed Control

The proposed approach assumes that the MDC path is constituted by more-or-less circular rounds, or laps, starting from the base station and eventually coming back to it. It assumes straight lines inside the communication range of each node. The target was to adapt the MDC speed while visiting static nodes for data collection. Before presenting the proposed adaptive speed control policy for MDC, it is necessary to define the parameters that may affect the performance of data collection in such scenarios.

The first and most important parameter is the amount of sojourn distance (actual contact distance) of MDC with each CH in the area of interest. The length of this distance is important in designing adaptive speed control for MDC since the actual transfer of data between CH and MDC occur at this distance.

The second parameter is the amount of data to be collected from each cluster or static node. This parameter is crucial in deciding the optimum speed for MDC during the sojourn distance with each cluster head. Optimum speed can be defined as the speed of the MDC that results in maximum data collection rates with minimum possible traverse time.

The third parameter is the quality of communication link with each of the cluster or static node lying on the MDC trajectory. This parameter allows a fine-grain tuning of the MDC motion. It must be noted here that the quality of communication link with each of the cluster or static node can also be calculated well in advance if the sojourn distances and locations of the corresponding static nodes are known.

The fourth parameter is the relation between data collection rate and MDC speed in the given environment. This parameter is equally important in deciding a favorable MDC speed during data collection. Such relation can be learned at runtime or can be provided at design time if the network conditions and environment are known in advance.

Once these control parameters are estimated, it is then possible to find a relation to calculate an optimum speed for the MDC. Finally, an adaptive algorithm is used to periodically check the validity of this optimized speed and fine tune it if necessary, in order to meet the changing environment throughout the data collection round.

3.3.2.1 Sojourn Distance

The actual *sojourn distance*, or *sojourn time*, of the MDC with the static source nodes, is an important factor and can be calculated well in advance if the trajectory of the mobile node and the location and communication range of the corresponding static nodes are known. If the trajectory of the MDC remains fixed (for instance, in the case of a shuttle), calculation of *sojourn distance* is required only once in the beginning of the first data collection round. If the MDC trajectory is dynamic and changes with time, the sojourn distances with static nodes can be estimated on the fly using MDC motion path and location of static nodes.

Consider the MDC is moving towards static nodes with speed v_{mdc} on a straight line. For simplicity, the path of the MDC is assumed to be a straight line within the communication range of a CH. In this case, the path of the MDC can be represented by Equation 3.1 where $\tan\theta$ is the slope of the line, x_{mdc} and y_{mdc} are the coordinates of the MDC position, and c is the y-intercept.

$$y_{mdc} = x_{mdc} \tan\theta + c \tag{3.1}$$

The sojourn distances of an MDC to nodes i, j, and k are represented by d_i, d_j, and d_k respectively in Figure 17. As it can be observed in the Figure 17, the sojourn distance d_k is larger than the distance d_i. Moreover, the sojourn distance d_j is zero, since the MDC trajectory does not lies in the communication range of node j. In addition, *sojourn time* with node k is the time that the MDC takes to travel distance d_k. In other words, it is the amount of time that the MDC is within the communication range of node k.

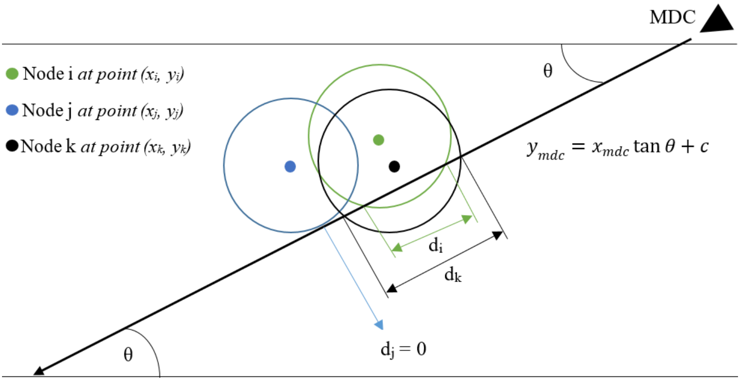


Figure 17 – Sojourn distance of MDC with static nodes

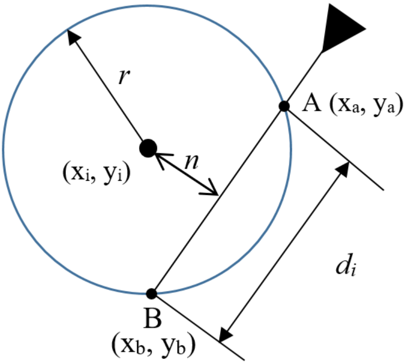


Figure 18 – Estimating Sojourn distance of MDC with static nodes

To exemplify, let's consider that the MDC enters into the communication range of node i at point A and exits at point B, as illustrated in Figure 18. Here, d_i is the distance between points A and B, and r is the radius of the communication range of node i. Using simple geometry techniques, it is possible to find that:

$$(x_a - x_i)^2 + (y_a - y_i)^2 = r^2 \tag{3.2}$$

If the location of node i is known, then by combining Equations 3.1 and 3.2, one can easily find the intersection points A and B as:

$$x_{a,b} = x_i \pm \sqrt{r^2 - (x_{mdc} \tan\theta + c - y_i)^2} \tag{3.3}$$

$$y_{a,b} = x_{a,b} \tan\theta + c \tag{3.4}$$

After finding the coordinates of points A and B, the sojourn distance d_i of MDC with node i is calculated using Equation 3.6. The $sojourn\, time_i$ of MDC with node i is obtained through Equation 3.7.

In the case of an aerial MDC, which moves in the sensing field with an altitude of h above the ground level (sensors are assumed to be deployed on the surface of the ground), the **effective sojourn distance** is smaller than the one calculated on ground. Considering the communication zone of a sensor node in space as a sphere, previously shown in Figure 5 and repeated here as Figure 19, the **effective communication radius** (r_e) in the upper hemisphere is smaller than the **original communication radius** (r_o) at ground level. If the MDC is moving at an altitude of h above the ground sensor level, then the effective communication circle for the MDC is reduced, and the same for the entry and exit points and sojourn distance. In this case, r in Equations 3.2 and 3.3 should be replaced with r_e in order to accommodate the MDC height. The value of r_e can be calculated from Equation 3.5 using the popular Pythagoras theorem. If the value of h is zero, r_e become equals to r_o. It must be noted that the altitude of the aerial MDC inside the communication zone of each CH is assumed to be constant.

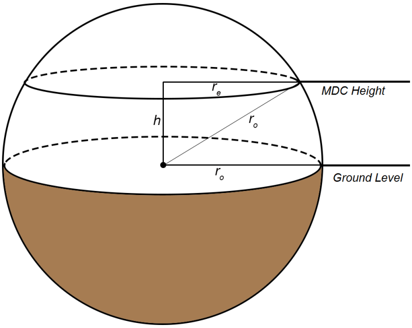


Figure 19 – [Figure 5 repetition]: 3d communication range of a ground sensor node

$$r_e = \sqrt{r_o^2 - h^2} \tag{3.5}$$

$$d_i = \sqrt{((x_b - x_a)^2 + ((y_b - y_a)^2} \tag{3.6}$$

$$sojourn\, time_i = \frac{d_i}{v_{mdc}} \tag{3.7}$$

In this way, knowing the location of all static nodes and their communication range, the MDC can estimate its corresponding sojourn distance using Equation 3.6 and sojourn time using Equation 3.7.

3.3.2.2 Data to be Collected from Static Nodes

The MDC needs to adjust its speed according to the total amount of data that each CH wishes to transmit. If the network's operation parameters (data generation and processing) is fixed, the total amount of data each cluster is producing in a given time can be estimated well in advance. If the network is dynamic, this information can be obtained from each CH, whenever it enters its communication range. In reply to the MDC beacon message, each CH informs the MDC regarding the total number of data packets it wishes to transfer. The MDC stores this information and uses it in its optimum speed estimation.

3.3.2.3 Relation between MDC Speed and Data Collection Rate

The speed of mobile data collector may influence the data collection process in two ways. First, the MDC speed considerably increases or decreases the actual transfer time (sojourn time) with the static node, thereby greatly influence the performance of data collection. However, this effect has already been taken into account when the sojourn time was calculated using Equation 3.7.

Second, the speed in itself may affect the reception of a wireless communication from the corresponding CHs. Considering that at speed v the MDC collects N packets while remains in contact with a CH for t time. This brings the question of what will be the number of packets collected if the MDC remains in contact with a CH for the same t time but moving with $2v$ or any other speed. In other words, if the MDC speed is increased or decreased, it is necessary to evaluate its effect on the rate of packets (packets/sec) received by the MDC.

For the moment, it is assumed here that the number of packet per seconds changes linearly with increasing the MDC speed. This relation is denoted as β, which is the rate of change of data rate with respect to MDC speed and can be calculated using Equation 3.8.

$$\beta = \frac{data\,rate_{new} - data\,rate_{old}}{v_{new} - v_{old}} \tag{3.8}$$

3.3.2.4 Link Quality

It must be noted that the quality of the communication link between each CH and the MDC is also important, given that it affects the data collection process. The quality of the communication link is directly related to the distance between sender and receiver. The larger the distance between the MDC and CH, the worse is the quality of the communication link. Link quality is degraded by factors such as the distance between sender and receiver, multipath propagation, interference, and hardware transceivers (BACCOUR et al., 2013). However, the distance between sender and receiver is the prominent cause and is used in predicting the quality of data collecting in this scheme.

Consider the distance between the midpoint of chord AB (Sojourn Distance) and the CH position to be called n (see Figure 18). Here n can be used as a fairly simple estimate of the best link quality that can be obtained between MDC and CH. It is the minimum distance during the whole contact time between CH and MDC. For instance, moving in a straight line exactly over the CH ($n = 0$) gives the theoretical maximum possible link quality between MDC and CH (let us call it LQ_0). Similarly, moving in a straight line away from CH decreases the quality of the communication link (let's call it LQ_n). The maximum link quality is represented by $LQ_0 = 1$ (when $n = 0$) and worse link quality by $LQ_n = 0$ (when $n = r$), depending on the distance between chord AB and CH location. After calculating the sojourn distance, the MDC can easily calculate n with each of the CHs using Equation 3.9 and its corresponding link quality using Equation 3.10. Equation 3.10 can be regarded as a link quality indicator which can be used for calculating the optimum speed for MDC.

$$n^2 = r^2 - (\frac{\overline{AB}}{2})^2 \tag{3.9}$$

$$LQ_n = \frac{r - n}{r} \tag{3.10}$$

In the case of aerial MDC which moves in the sensing field with an altitude of h above the ground level, the **effective separation** (n_e) between sensor node and center of MDC path, is greater than the n (one calculated for ground-based MDC). Consider the communication zone of a sensor node in space as a sphere shown in Fig 20, where the effective distance between the ground sensor node and center of MDC path is depicted as n_e. In this case, n in Equation 3.10 should be replaced with n_e in order to accommodate the height of MDC. The value of n_e can be calculated from the Equation 3.11 using the popular Pythagoras theorem. If the value of h is zero, n_e become equal to n.

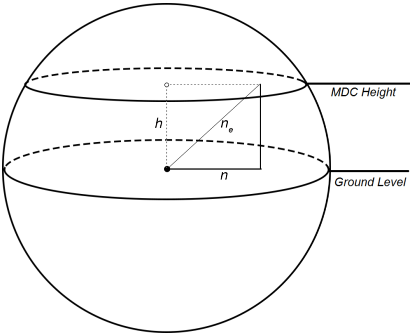


Figure 20 – 3d communication range of a ground sensor node with a diagonal value of n

$$n_e = \sqrt{n^2 + h^2} \tag{3.11}$$

3.3.3 Optimum Speed Calculation

While on a mission for data collection from a WSN, the MDC will be in one of the three states presented in Figure 21. The MDC starts its mission by switching to the *approaching* state. In this state, the MDC is not collecting any data and is trying to reach the next static node for data collection and therefore has the liberty to move at any speed within its range. In this case, the MDC can be either set to move at its maximum speed or application defined favorable speed for traversing (call it v_{fav}). As soon as the MDC reaches a point within a reasonable range of (say m meter) the communication zone of its next possible data collection point, it goes to *connecting* state. In this state, the MDC starts transmitting beacon messages, waiting for a connection to be established with the data collection point. If this is a first attempt to establish a connection with a static node (the previous state is *approaching*), the MDC tries to achieve an application defined favorable speed for data collection (call it v_{data}) since it is highly probable that in few moment the MDC will start collecting data from the corresponding static node. If MDC switches to *connecting* state from *collecting* state, it moves according to the optimum speed calculated through algorithm 2. During the connecting state, as soon as a beacon reply is received, if the static node has data to transmit, the MDC goes to *collecting* state, otherwise it goes to *approaching* state. In *collecting* state, the MDC starts receiving data packets from the static node while at the same time estimate, achieve and maintain an optimum speed for its motion. In *collecting* state, if MDC reach a point m meter away from the communication zone of its next possible data collection point, it does not change its state (already collecting data) while it either does not change its speed or moves with v_{data} (whichever is minimum). MDC goes to *approaching* state whenever it goes out of the communication range of the static node or data collection is over with the corresponding static node.

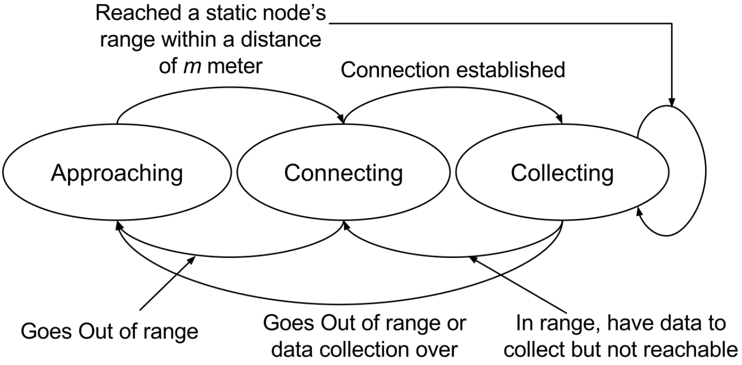


Figure 21 – State Diagram for MDC Mission

Knowing the data collection rate ($data\,rate_{old}$) of MDC at speed v_{old} and the value of β in a given environment, it is possible to estimate the data collection rate ($data\,rate_{new}$) at speed v_{new} using Equation 3.12. It must be noted that $data\,rate_{old}$ is assumed to be the number of packet received per second by the MDC with link quality LQ_0. Similarly, $data\,rate_{new}$ is also the data rate with link quality LQ_0. However, depending on the value of n, the $data\,rate_{new}$ with link quality LQ_n can be calculated using Equation 3.13.

$$data\,rate_{new} = data\,rate_{old} + (v_{new} - v_{old})\beta \tag{3.12}$$

$$data\,rate_{new} = LQ_n(data\,rate_{old} + (v_{new} - v_{old})\beta) \tag{3.13}$$

Similarly, the number of packets N_{new} estimated to be received using v_{new} is:

$$N_{new} = \frac{(data\,rate_{new})d_i}{v_{new}}$$

where d_i is the *sojourn distance* of MDC with node i. Rearranging and putting the value of $data\,rate_{new}$ from Equation 3.12 gives:

$$v_{new} = \frac{d_i(data\,rate_{old} + \beta v_{old})}{N_{new} + \beta d_i} \tag{3.14}$$

After approaching a CH and knowing the value of d_i, the number of packets to be collected, the MDC can estimate its optimal speed using Equation 3.14. Similarly, depending on the value of n, v_{new} with link quality LQ_n can be calculated as:

$$v_{new} = \frac{LQ_n d_i(data\,rate_{old} + \beta v_{old})}{N_{new} + \beta d_i} \tag{3.15}$$

The seed values of $data\,rate_{old}$, v_{old}, and β in Equation 3.14, can be made available to the MDC at design time or can be learn at runtime. If MDC has no information about v_{old} and β, or if the value of β is zero or negligible, then Equation 3.15 reduces to Equation 3.16.

$$v_{new} = \frac{LQ_n d_i(data\,rate)}{N_{new}} \tag{3.16}$$

Where $data\,rate$ is a general value of data collection rate in the given sensing environment irrespective of MDC speed (could be for any MDC speed). In this case, the optimum speed is calculated only taking into account the sojourn time with MDC. This general value of data collection rate gives the MDC a clue about the sensing environment and data collection process. In the case of no information about the $data\,rate$ in a given scenario, some blind value for v_{new} can be selected, where MDC can gradually converge to an optimum speed in the due course of time. It must be noticed, however, that the MDC can only change speed according to its allowed acceleration and deceleration capability. If for example, one CH allows the MDC to collect data at maximum speed and a following close neighbor requires minimum speed, it is probable that the MDC will not be able to maintain an optimum speed and collect all data from the second CH with minimum speed, as it might not have enough time to decelerate until reaching the second node. Same will be the case, if MDC is in *approaching* state moving with a maximum speed and the next data collection point require minimum speed for data collection. However, this situation can be avoided or minimized by properly selecting the values of m and v_{data} described earlier in this section. This will ensure a fast moving MDC to start applying brakes at a distance of m meter before reaching communication zone of its next possible data collection point and will minimize wasting valuable contact time.

With each subsequent attempt of data collection, the MDC saves the record of its current speed versus number of successful packets received and uses it as a seed value in optimum speed prediction in its future data collection laps.

It is believed that in a given scenario, the data collection rates, and data collection delays can also be influenced by factors other than speed. But speed is the most significant factor because it greatly varies the amount of time for which the MDC and CH can remain in contact.

3.3.3.1 Maintaining MDC Optimum Speed

Irrespective of using Equation 3.15 or 3.16 or some blind value for optimum speed calculation, it is important to verify and maintain the correctness and suitability of MDC speed continuously. Data collection rates and wireless communication reception may vary in different situations and the selected optimum speed may not be the best speed in the current circumstances.

Consider an MDC selects an optimum speed and starts receiving data from a CH. After some time, the remaining number of packets to be received from current CH is N_{rem}. Current speed and data collection rate of MDC is v_c and $data\,rate_c$ respectively and remaining sojourn distance with current CH is d_{rem}. MDC uses algorithm 2 to continuously check the validity of its speed and make any amendment or recalculate its optimum speed based on the new parameters if necessary.

Algorithm 2: Maintaining Optimum Speed

```
while in communication range with current CH do
    Find $N_{optimum} = (d_{rem} data\,rate_c)/v_c$ after every $t$ seconds (default 1s);
    if $N_{optimum} = N_{rem}$ then
        Current $v_c$ is good;
    else if $N_{optimum} > N_{rem}$ then
        Increase $v_c$ by some factor or recalculate $v$ for MDC;
    else if $N_{optimum} < N_{rem}$ then
        Decrease $v_c$ by some factor or recalculate $v$ for MDC;
    end
end
```

3.3.3.2 Time Bounded Data Collection

Adaptive speed control can be very useful particularly in time bounded data collection. Suppose the maximum time required for sensors to transmit its data is T. This means that the maximum time the MDC can spend in one round across the sensor network is also equal to T. In this case, the goal is to maximize the amount of data collected by MDC in one round. A simple approach for the MDC to move is to calculate v using T and the length of the MDC trajectory. But using adaptive speed control the MDC can perform better than using this simple approach. Data collection in a time constraint scenario is increased if the MDC moves slowly when it is busy in data collection and faster if there is no node in range (particularly in the case of sparse networks).

The total amount of data collected in one lap can be estimated, knowing the path and speed of mobile nodes and location of all static nodes as:

$$Data = \sum_{i=1}^{Z} [(LQ_n^i)(sojourn\,time_i)(data\,rate_i)] \tag{3.17}$$

Where Z is the total number of cluster heads, LQ_n^i is the link quality of MDC with node i, and $data\,rate_i$ is Node i data rate.

3.4 PERFORMANCE EVALUATION

The evaluations presented in this section were done by means of simulations using Omnet++ 4.6 tool with Inetmanet-2.0 framework. All nodes use radio model IEEE 802.15.4.

3.4.1 Evaluating β

Before analyzing the data collection process, it is here described an attempt to estimate the value of β. Therefore, the MDC is placed at a point such that it lies well in the communication range of a static node and then it is moved in a circle with different speeds, as shown in Figure 22(a). The distance between MDC and the static node is kept constant while location and speed of MDC are changed. The static node is set to continuously transmit packets so that the amount of data collected depends essentially on the time for which the MDC is in range. The MDC counts the number of received packets. This experiment is performed for different speeds (from 0m/s to 30m/s) and different propagation models (*TwoRayGroundModel*, *RiceModel*, *RayleighModel*, and *LogNormalShadowingModel*). The results for *LogNormalShadowingModel* are plotted in Figure 23 (blue line). It was found almost no difference in the number of packets collected per second at all speeds of the MDC. This result is consistent for all the propagation models used during the simulations. This behavior is also confirmed by (KANSAL et al., 2004b) for a mobile node having speed ranging from 0.5m/s to 2m/s. This shows that the speed alone has negligible influence on the number of packets received per second by the MDC.

In the second case, the MDC is moved from one edge of the communication circle of the CH to another on a straight line, as shown in Figure 22(b). Again, this experiment is performed for a number of different speeds, ranging from 0m/s to 30m/s, with different values of n (separation between MDC and CH). The results are plotted in Figure 23 (orange line). This time, the number of packets collected by MDC per second slightly decreased with increasing the MDC speed. This slight variation in data collection is due to the fact that the distance between the static node and MDC changes during data collection time. The closer the distance the better the reception of packets and more the number of successful transmission.

The analysis of MDC speed versus data collection rate suggests that the data transmission when the MDC is within a good communi-

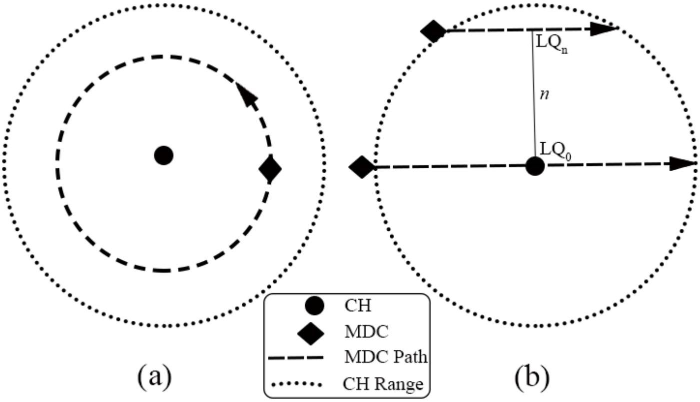


Figure 22 – Studying influence of different MDC speeds (a) in circle (b) on a straight line

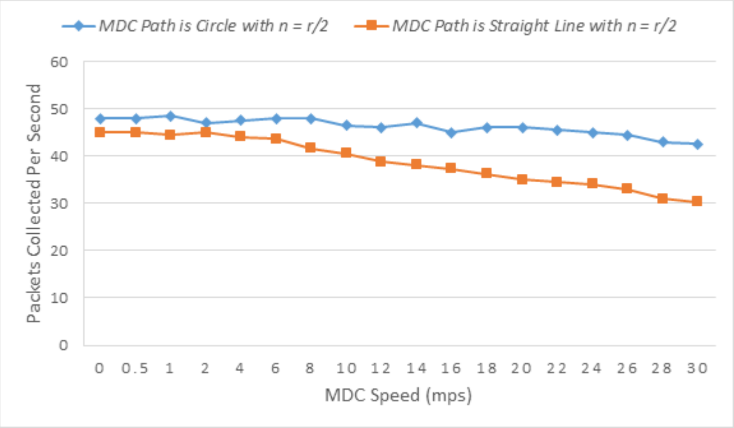


Figure 23 – Data Collection Rate at different MDC speeds for network typologies shown in Figure 22

cation range of the static node does not depend heavily on the speed alone, but on the sojourn time with MDC.

From this simulation experiment (orange line in Figure 23) it is roughly estimated that for every 1m/s increase in the MDC speed there is a 0.5 packets/sec decrease in the data collection rate. So using

Equation 3.8, β turn out to be 0.5.

3.4.2 Evaluating Link Quality

To evaluate LQ_n, the MDC was moved from one edge of the communication circle of the CH to another on a straight line with different values of n (separation between MDC and CH). The results are plotted in Figure 24 where the value of LQ_n decreases with increasing the value of n. This is due to that fact that the more the separation between MDC and CH, the lower the quality of the communication link between them.

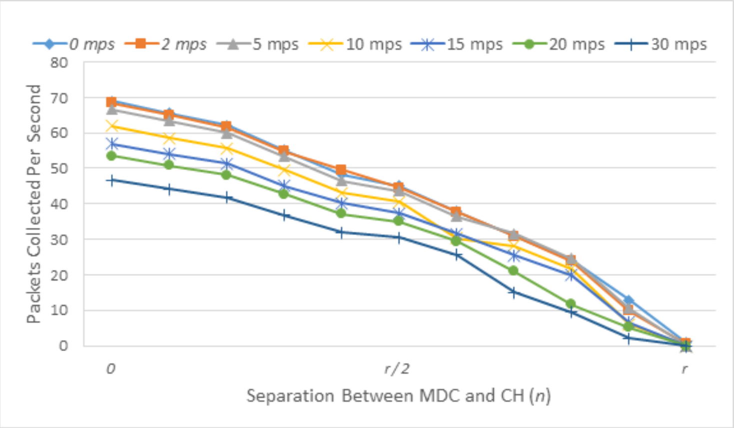


Figure 24 – Data Collection Rate at different MDC speeds with different n

Figures 25 and 26 show a comparison between theoretical and simulated behavior of data collection at different MDC speeds. Similarly, Figure 27 presents the results of practical experiments performed in a related Master Thesis (BODANESE et al., 2014), where the MDC speed is fixed at 7m/s and different values of n are tested. Observing the simulated and practical experiments it can be seen that the decrease in data collection with the increase of n is not so strictly linear as predicted by Equation 3.10. However, the Equation is close enough to be used in speed optimization.

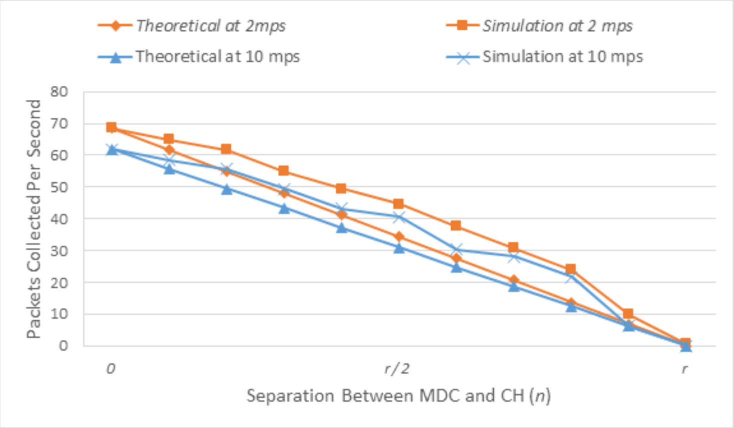


Figure 25 – Theoretical vs Simulated Data Collection Rate at 2 and 10m/s with different n

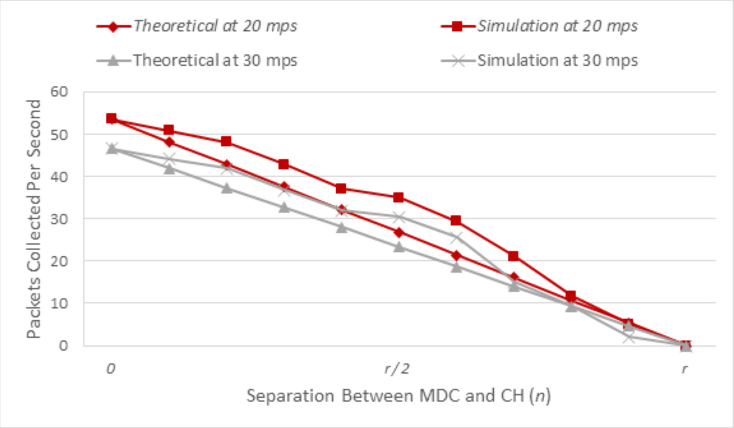


Figure 26 – Theoretical vs Simulated Data Collection Rate at 20 and 30m/s with different n

3.4.3 Data Collection

To validate the feasibility of the proposed scheme, an 800m by 800m simulated environment of a WSN scenario is developed. It includes the reasonable assumptions that MDC knows the location of

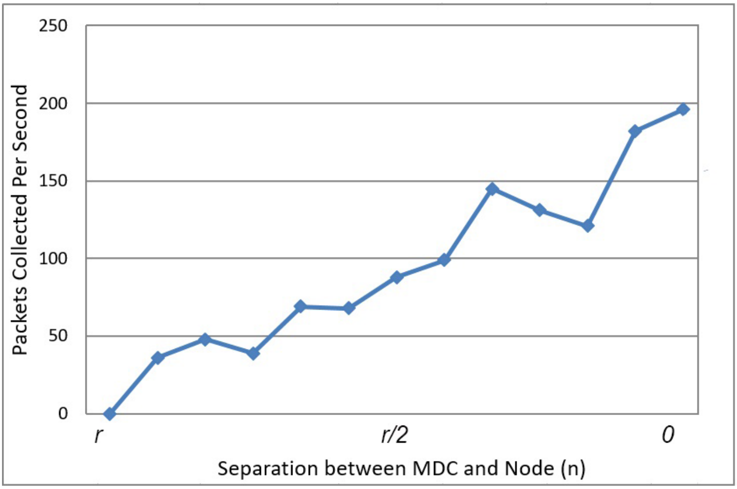


Figure 27 – Practical Data Collection Rate at the MDC speed of 7m/s with different n (Figure adapted from (BODANESE et al., 2014))

each CH in the area of interest and that it also can calculate its own location during data collection process. The MDC path is assumed to be a straight line while passing through each cluster zone while the propagation model used in the simulation is free space.

Figure 28 shows the simulation scenario, where 50 variable sampling nodes (with variable data generation capability) are randomly deployed and organized into 6 clusters on the basis of the distance from each other. The transmission range of each CH is approximately a circle with an 80m radius. The length of the track and hence each data collection lap is 1900m approx. The acceleration and deceleration capability of MDC is assumed to be $\pm 2m/s^2$. Values for m, v_{fav} and v_{data} for MDC is defined to be 15m, 15m/s and 5m/s respectively. As soon as the mission begins, the MDC starts traversing the network area on the pre-defined track and starts collecting data by periodically sending a beacon message. When a CH receives this beacon it replies with the total number of data packets it wishes to transfer. The MDC calculates the sojourn distance and link quality distance (n) with each CH reachable from its mission trajectory and estimates its optimized speed with the aim to grab all the remaining data from each CH in one pass. Any leftover data with a CH is collected in the next lap.

It is assumed that a sensor reading takes 2 bytes of storage in

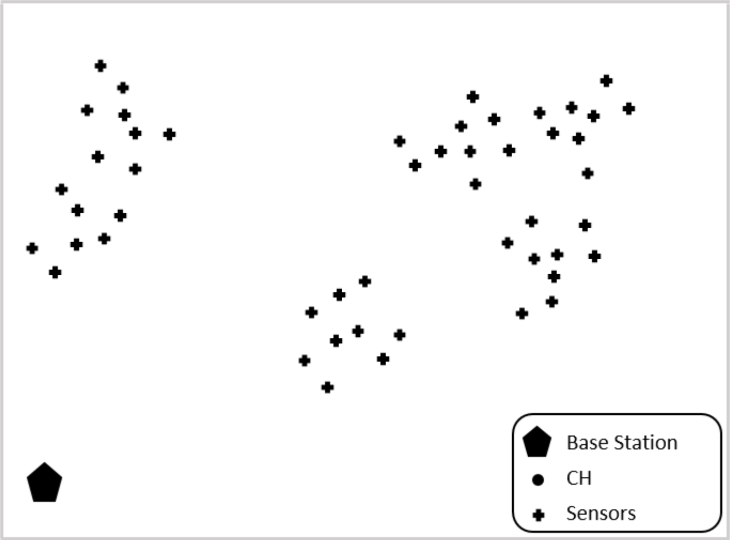


Figure 28 – WSN Scenario for adaptive speed control

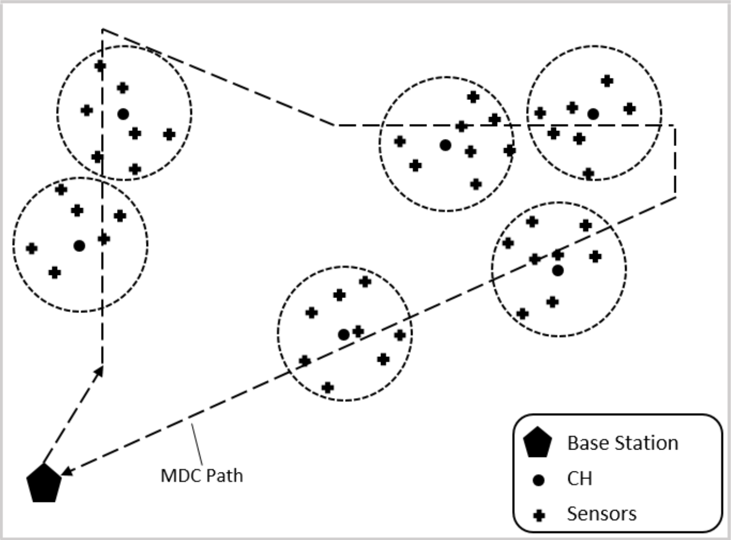


Figure 29 – WSN Scenario for adaptive speed control with MDC Path

the memory. Each 20% of nodes take 60, 20, 6, 2, and 1 reading per minute. These variable sampling nodes are scattered randomly with the sensing area. This generates 34,176 packets of data in 24 hours when each packet has 75 bytes. It is considered that the MDC is used after each 24 hours to collect all these samples for further processing. In this case, each CH has a different number of packets to be transmitted when the MDC approaches it.

Figs. 30, 31, and 32 present the comparison of the data collection process when the MDC moves at uniform speed (up to 30m/s) vs adaptive speed.

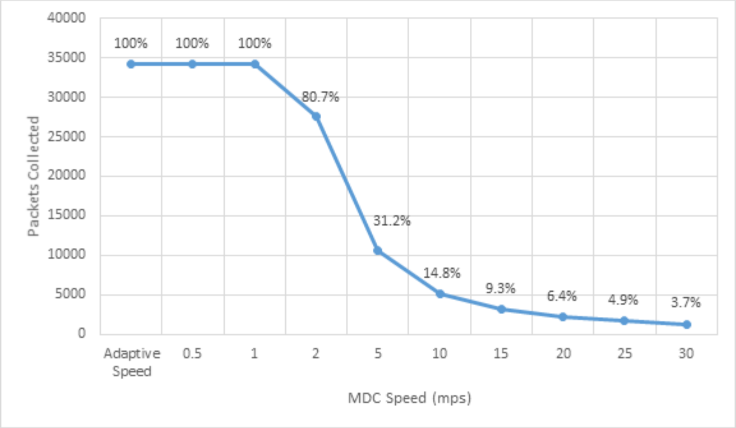


Figure 30 – Data collected in one round with different MDC speeds

Figure 30 shows the number of packets collected and percentage of total data collected in one MDC round. The amount of data collected by the MDC decreases with increasing the MDC speed since at higher speeds the actual contact time between MDC and CH is decreased. It must be noted that the slower the MDC moves, the longer the mission takes.

Figure 31 shows the number of laps (MDC rounds) required to collect all generated data with different MDC speeds. At higher speeds, a greater number of laps are required to gather all the data. This is due to the fact that at high speeds the MDC and underlying CH get less time for data transfer (contact time) and, as a result, the MDC needs more laps to finish the collection of all data available.

It can be noticed in Figure 32 that at higher speeds the MDC requires less amount of time to complete one round, but at the same

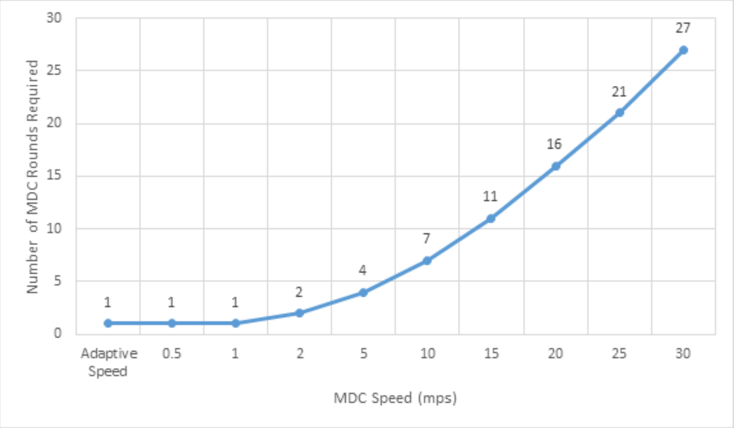


Figure 31 – MDC rounds required for collecting all sensors data

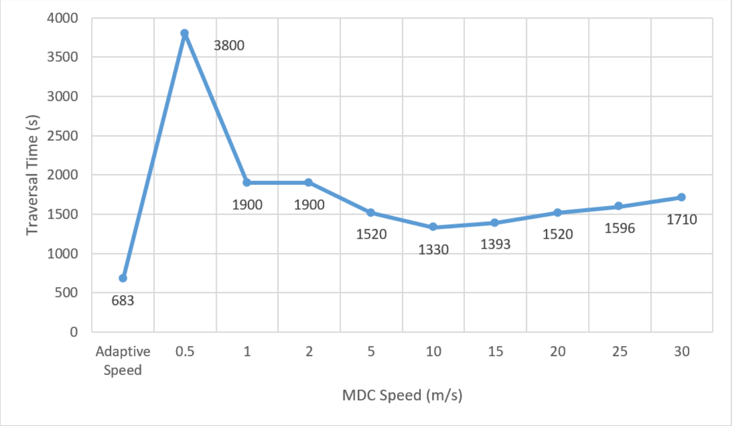


Figure 32 – Total MDC traversal time required for collecting all sensors data

time, a greater number of laps to grab all data is required.

It verifies that at uniform speed the MDC cannot provide optimum results. There are areas where the MDC needs to move faster as well as areas where the MDC needs to move slower. So best performance both in terms of number of laps and traversal time can be obtained only with adaptive speed control. It is the adaptive speed scheme which is best in terms of MDC traversal time, the number of laps, and data collection rates. It can also be noticed that in the case of adaptive speed, traversal time and data transfer delays are minimized by up to 80%. The proposed adaptive speed control performs better particularly in the case of large and sparse sensor networks or heterogeneous sensor networks. In sparse networks, devices are deployed too far apart and the network is split into multiple fragments. There exist segments on the MDC path where there are no nodes in range, so the MDC is allowed to move faster up to its limit (or a pre-defined best value) to save time. Similarly, in heterogeneous networks, the data collection requirement is also not uniform throughout the sensing area and hence the MDC has the freedom to control its motion.

3.4.4 Energy / Fuel Consumption of MDC

The energy / fuel consumption of MDC during traversing the network was not directly measured from the simulated experiments since this measurement mostly depend on the nature, type, weight, volume, and speed of the mobile node as well as on other different factors. There are large variations in the operating conditions and some factors are difficult to simulate (TONG; HUNG; CHEUNG, 2000). However, it can be deduced that the energy consumption of mobile nodes is directly related to their traverse time. As stated in (TONG; HUNG; CHEUNG, 2000), four standard driving modes can be defined for ground vehicles, as follows:

- **Idling Mode** where the vehicle has zero speed and zero acceleration.
- **Acceleration Mode** where the vehicle has a positive incremental speed of more than $0.1 m/sec^2$ during 1 sec interval.
- **Cruising Mode** where the vehicle has absolute incremental speed changes of less than or equal to $0.1 m/sec^2$ during 1 sec interval.
- **Deceleration Mode** where the vehicle has speed decrease of more than $0.1 m/sec^2$ during 1 sec interval.

Authors in (TONG; HUNG; CHEUNG, 2000) claim that the *cruising mode* of ground vehicles consumes approximately 50% less fuel than *acceleration mode*, while *deceleration mode* consumes approximately 60% less.

Consider a scenario where a vehicle moves at a uniform speed for 60 minutes while consuming F fuel per minute. In this case, the total fuel consumption for the trip will be $60F$. Now consider the vehicle accelerates for 10 minutes, decelerates for 10 minutes, and cruises for 40 minutes. Roughly speaking, the total amount of acceleration and declaration are almost equal for a complete trip of a vehicle performing any task. In the latter case, the total fuel consumption will be $(2F * 10) + (0.9 * F * 10) + (F * 40) = 69F$. Similarly, the fuel consumption roughly increases to $87F$ (with a maximum of 150% in total) if for 30 minutes the vehicle is in acceleration mode and for the remaining time it is in deceleration mode.

Figure 33 shows the pattern and variations of MDC speed versus traverse time during adaptive speed. It can be seen that the MDC adjust its speed according to the data collection requirements of the network.

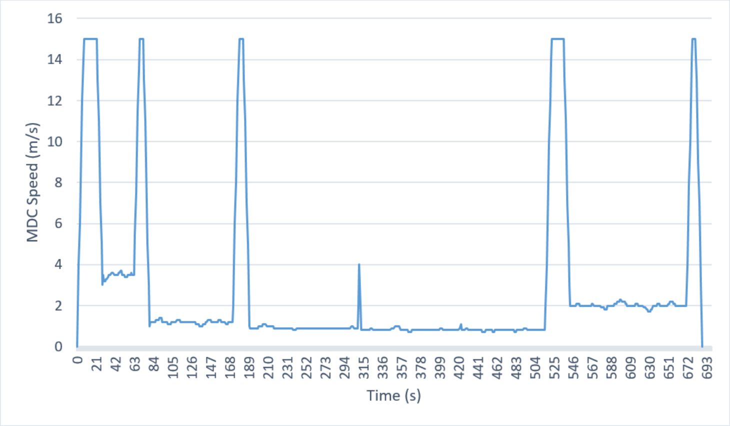


Figure 33 – MDC speed versus traverse time in adaptive speed case

Observing Figure 33 it can be deduced that the MDC in the simulation experiments remains in acceleration and deceleration mode for about 20% of its traverse time and in cruise mode for the remaining 80% of the time in case of the adaptive speed control. In this case, an estimation of the fuel consumption of MDC for collecting all sensor data can be made, considering $2F$ fuel to be consumed per sec in acceleration mode and F fuel to be consumed in deceleration and cruise mode. Figure 34 shows a rough estimation of the total fuel consumption of MDC while traversing the network and collecting all the generated data. Figures 34 and 32 have very similar pattern, since at uniform speed, the fuel consumption of MDC is essentially proportional to its traverse time. However, in adaptive speed control, it also depends on the variation of MDC speed with respect to time. It can be seen that despite the higher fuel consumption during acceleration mode, adaptive speed still outperforms uniform speed in terms of fuel consumption too.

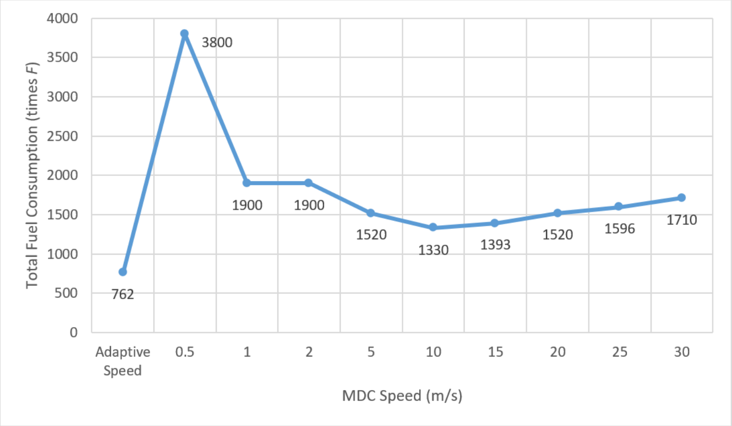


Figure 34 – A rough estimate of the total fuel consumption of MDC while traversing the network for data collection

3.5 CHAPTER CONCLUSIONS

This chapter investigated and demonstrated the viability and usefulness of dynamic speed control for MDC while collecting data from a wireless sensor network. The proposed implementation design saves significant MDC traversal time and number of laps while increasing data collection rates and minimizing data transfer delays. The scheme is particularly advantageous for large, sparse, and heterogeneous sensor networks. The approach proposed for modifying the MDC speed is capable of adapting to changes in data collection requirements.

4 MOBILITY AWARE SLEEP WAKEUP SCHEDULING

Maximizing network lifetime and improving data collections rates are among the most important issues in wireless sensor networks. Efficient energy management schemes must be utilized at sensor nodes to prolong the network lifetime (KANSAL et al., 2007; WILLIG, 2008). Similarly, data collection, which involves the transmission of data from source nodes to a base station, is the most important operation in WSNs. Sensors data must be efficiently transmitted towards a central base station or sink in a timely manner.

Mobile elements are often introduced in WSN to play a number of different roles. In this regard, the most important role is a mobile data collector. Mobile data collector (MDC) is a mobile node responsible for performing a specific task of collecting data from sensor nodes when in their coverage range. They are assumed to be powerful in terms of data storage, energy, and processing capabilities and are usually equipped with specialized hardware such as GPS units etc. An MDC can serve either as a Mobile Sink (MS) by consuming the collected data itself or as a Mobile Relay (MR), which collect from sensors and eventually transfer it to the base station whenever possible. In either case, the MDC moves to the different regions of the network with random, predictable, or controlled mobility, and performs the task of data gathering and /or dissemination.

Data collection in WSNs using an MDC has many advantages if compared to traditional multihop data collection schemes. First, the network doesn't need to be connected and dense deployment of sensor nodes is not a requirement. Second, the network operation is simplified and reliability is increased as the MDC directly collects data from sensor nodes. Finally, WSN lifetime might be extended by minimizing and spreading the energy consumption more uniformly throughout the network.

However, data collection with MDC introduces new challenges. The most important among them is the efficient and timely discovery of MDC by static nodes and hence maximizing data collection during the available contact time. Mobile data collectors may have a very short connection time to other sensor nodes in WSNs. An MDC moving with a speed of 5m/s will be within a radio range (50m) of a sensor node for at most 10s. During this short period of time, the sensors have to detect the MDC, establish a connection, and exchange the application data. In data collection applications, the discovery and data transfer to MDC need special attention since the actual contact time with MDC is always a scarce resource in such environments.

Data communication between sensor nodes and MDC takes place in two phases. First, sensor nodes need to discover the presence of MDC in their communication range. Once discovered, sensor nodes can then transfer their data to the MDC. Discovery of MDC and data transfer has to be energy efficient and collision free in order to prolong network lifetime. In addition, these two phases should facilitate maximum data transfer between MDC and sensor node in a reliable and efficient way. The first challenge here is the discovery of the mobile node since sensor nodes usually do not have a priori knowledge of the MDC mobility pattern. For this to happen, the sensor node should be in listening state (in discovery mode) as soon as the MDC enters into its communication range. The more the sensor node takes time in discovering and recognizing the MDC, the less sojourn time remains for data collection.

In addition, collision should be minimized during the data transfer between MDC and sensor node in order to improve data collection process. Hence, dynamic resource management schemes can play a key role in successful data collection applications (HADIM; MOHAMED, 2006). Flexible and adaptive solutions are thus needed to adjust the sleep/wakeup periods and medium access, depending on the mobility patterns of the mobile data collector.

Three major issues have to be tackled in increasing performance of WSN in terms of energy and data collection rates. First, collision and retransmission attempts must be minimized. In the case of message collisions, data has to be retransmitted, increasing energy consumption and degrading the data collection process. Collisions and retransmissions increase delays and decrease data collection rates as well. Second, overhearing has to be avoided. Overhearing means picking up packets that are intended to other nodes. Third, idle listening has to be minimized. In the idle listing, a node listens to receive possible traffic that is not there. Many measurements (STEMM et al., 1997) have shown that idle listening consumes above 50% of the energy required for receiving.

The transmission system of a sensor node typically consumes much more energy than the sensing and processing components. In addition, while being idle, the radio transceiver consumes approximately the same power as in receive modes (RAGHUNATHAN et al., 2002). On the other hand, it consumes more power in transmit mode and significantly less power in the sleep (low power) mode. Therefore, the most simple and effective approach to conserve energy is duty-cycling, which consists of putting the radio in sleep mode during idle periods. Sensor nodes alternate between sleep and wake-up periods, and they have to coordinate their sleep schedule in order to make communication feasible and efficient (GANESAN et al., 2004). To maximize the network lifetime and data collection process, protocols usually put sensors in a sleep mode for most of the time and only let them wake up periodically for data communication. Moreover, proper medium access control should be defined in order to avoid or minimize data collisions and retransmissions attempts by sensor nodes.

This chapter presents the Mobility Aware Sleep Wakeup Scheduling (MASWS), focusing on the problem of MDC discovery and medium access control in a WSN that uses an MDC to collect data. The protocol exploits an adaptive time division multiple access (TDMA) scheme which is adjusted according to the mobility patterns of MDC, achieving both low energy consumption and high data collection. The protocol is targeted to data collection applications (e.g. monitoring and surveillance), in which sensor nodes have to periodically report to a sink node. Improving data collection rates and minimizing energy consumption was the primary goal of this design.

The protocol was implemented in Omnet++ and a detailed simulation analysis is carried out in order to demonstrate its effectiveness and measure its performance. The obtained results show that, thanks to its flexibility, MASWS outperforms the commonly used 802.15.4 beacon enabled scheme in terms of energy efficiency, MDC traverse time, and data collection rates.

The remainder of the chapter is organized as follows. Section 4.1 presents an overview of the related works, while Section 4.2 introduces the system model and proposed solution. Section 4.3 outlines the simulation setup and discuss performance evaluation and obtained results. Finally, Section 4.4 concludes the chapter.

4.1 RELATED WORKS

In recent years, a very large number of MAC and energy conservation schemes for WSNs have been proposed. The reader can refer to (ANASTASI et al., 2009; BACHIR et al., 2010; DONG; DARGIE, 2013) for a detailed survey on the most relevant proposals. In the following, some of the mobility-aware MAC protocols which support node mobility are explored.

The communication load distribution studied in (LUO; HUBAUX,

2005) shows that a network's lifetime can be improved even if an optimally placed fixed sink is replaced by a randomly moving mobile base station. The idea is to enable relay nodes to evenly consume energy and, thereby, optimize the overall energy consumption of the network.

In (SHAH et al., 2003b), mobile data collectors move randomly and collect the buffered samples opportunistically. The received data from the one-hop sensors are transferred to a wireless access point. Since the trajectory of mobile data collectors is random, the message transmission delay can be high.

A mobile data collector with predictable mobility pattern is used in (CHAKRABARTI; SABHARWAL; AAZHANG, 2003c) where static nodes are assumed to know the moving route of the mobile data collector, and this information is used to predict the time that data transfer may take place. Based on this predicted time and location, nodes schedule their sleep and listen periods to optimize its energy consumption.

A mobile data collector having controlled mobility with a heuristic solution called Earliest Deadline First is proposed in (SOMASUNDARA; RAMAMOORTHY; SRIVASTAVA, 2004b) to accommodate variable transmission rates. The aim is to actively control the movement of the MDC in real time. The node to be visited next by the mobile data collector is chosen as the one that has the earliest buffer overflow deadline. However, the approach does not work well if nodes with consecutive deadlines are located far away from each other.

The mobility-aware MAC protocol for sensor networks (MS-MAC) (PHAM; JHA, 2004) extends SMAC to support mobility. It introduces coordinated sleep/wake-up duty cycles and periodically synchronizes the schedule of nodes. The synchronization is done by broadcasting a SYNC packet at the beginning of the listen phase every predefined number of cycles (for example, 10 seconds every 2 minutes). A node first tries to follow the existing schedules by listening for a certain amount of time. If no SYNC packet is received, the node will randomly choose a time to go to sleep and immediately broadcasts this information. However, if a node receives a different schedule after it selects one, it will adopt both schedules. MS-MAC enables each node to discover the presence as well as the level of mobility within its neighborhood, based on the RSSI values obtained from the SYNC messages transmitted by its neighbors. If the RSSI value from one and the same neighbor changes during a time interval, it realizes that either this neighbor, the node itself, or both of them are moving, since a one-to-one mapping between the distance and the RSSI values is assumed. Depending on the change of the RSSI values, the relative moving speed of the mobile individual can be deduced. Based on this information, the node broadcasts a SYNC message containing its own schedule and additional mobility information (the maximum estimated speed in the neighborhood). Upon receiving this packet, all the neighbors create an active zone by adjusting the synchronization frequency if the node is to move from one virtual cluster to another. The synchronization frequency, however, depends on the maximum speed of the surrounding neighbors.

MMAC (ALI; SULEMAN; UZMI, 2005) is a mobility-adaptive, collision free, schedule-based MAC protocol. It introduces a flexible frame time that enables the protocol to dynamically adapt to mobility, making it suitable for wireless sensor environments. In MMAC, time is divided into rounds and each round is composed of k frames (k is an integer larger than 1). At the beginning of each frame, all the nodes in the network predict their mobility states at several different time points of the next frame based on the AR-1 mobility estimation model (ZAIDI; MARK, 2011). The average of these location estimations is regarded as a node's location prediction for the next frame. This information is transmitted to the node's corresponding cluster head. Since the cluster head never goes into sleep, it is able to collect the values of all its members and broadcasts them in the last slot of a frame. This ensures that all the nodes in the cluster have the best knowledge of the predicted mobility states of its current and potential two-hop neighbors. A node calculates the relative distance between the center node and itself, in order to learn whether it will enter or leave the cluster in the next frame. A node independently proposes a new frame duration and transmits it to the cluster head. The head, by averaging the duration estimations from all the members, produces the mean frame size and broadcasts it to all the nodes. If this value is less than the previous one stored at a node, it increases the random access interval and decreases the scheduled access interval while keeping the frame time constant.

Similarly, a mobility-aware TDMA-based MAC protocol for mobile sensor networks (M TDMA) (JHUMKA; KULKARNI, 2007) has been proposed to extend the TDMA mechanism for adapting to the changes in a network topology. Unlike a pure TDMA, M TDMA partitions the network into non-overlapping clusters using the FLOC algorithm (DEMIRBAS et al., 2006), with each cluster having its own head. Each node within a cluster is assigned a unique slot. To deal with mobility, some of these slots are shared across clusters and some of them are kept free for future allocation. To this end, M TDMA splits a given round into two parts, namely, the control part and the data part. The control part is used to adapt to mobility, whereas nodes transmit packets in the data part. Some of the slots at the end of the data part are reserved for the future entering nodes as well as the message retransmissions.

Authors in (ANASTASI; CONTI; FRANCESCO, 2009a) proposes an Adaptive Staggered sLEEp Protocol (ASLEEP) for efficient power management in wireless sensor networks targeted to periodic data acquisition. This protocol dynamically adjusts the sleep schedules of nodes to match the network demands, even in time-varying operating conditions. The scheme effectively reduces the energy consumption of sensor nodes (by dynamically adjusting their duty-cycle to current needs) under stationary conditions thus increasing network lifetime.

The light-weight mobility-aware medium access control protocol (MA-MAC) (ZHIYONG; DARGIE, 2010) is an extended version of XMAC. Similar to all the low duty cycle MAC protocols, MA-MAC enables a node to sleep most of the time and switches on the radio for receiving the incoming packets periodically. In the static scenario, MA-MAC performs similar to XMAC by dividing a preamble into multiple strobes and enabling an early ACK packet to save energy. However, if mobility is detected, MA-MAC initiates a seamless handover by relaying the remaining data to a new node before the link breaks. Each node can be found in one of the five states, namely, sleep, receive, send, discover, and handover. Initially, a node is in a sleep state, after being successfully booted. It may enter into a wake-up state if it has data to transmit or when its normal active period begins, or when a handover process is triggered. To support mobility, MA-MAC defines two distance thresholds. The first threshold prompts a node to initiate a seamless handover, whereas the second threshold sets an upper limit to the distance that should be traveled before the mobile node has established a link with a new relay neighbor. During mobility, if a transmitter detects that the distance between the receiving node and itself exceeds the first threshold, it enters into a discovery state and begins to search for an intermediate neighbor along the way to the base station. To do so, the transmitter broadcasts data packets in which handover requests are embedded. If it receives at least one ACK packet from a new node before it completes the second distance threshold, the transmitter enters into a handover state to resume data transmission to the newly discovered node. The transmitter enters into a sleep state otherwise.

The scheme proposed in (SHRESTHA; YOUN; SHARMA, 2010) called SWAP divides time into slots of equal length, and at the beginning of each slot, a sensor node enters either an active or power-saving state. The slot scheduling at each node is based on a binary vector, which is constructed using the mathematical properties of finite fields. The set of scheduling vectors generated by SWAP distributes active slots of nodes evenly over the entire time frame. This distribution of active slots reduces channel contention and allows better channel utilization. The SWAP scheduling scheme also ensures that any pair of neighboring nodes using the proposed scheduling scheme will be able to communicate each other since their active periods overlap at least once within a cycle of the sleep and wake-up slots. The proposed scheme also design a packet prioritization scheme in SWAP to reduce the packet latency of delay sensitive packets.

The mobile cluster MAC (MCMAC) (NABI et al., 2010) is a schedule-based MAC protocol which extends LMAC (HOESEL; HAVINGA, 2005) and GMAC (ANEMAET, 2008) to support cluster mobility. Unlike most of the proposed mobility-aware MAC protocol, MCMAC is optimized for those nodes which travel in a group. This is particularly the case in Body Area Networks, such as in healthcare applications, where a number of biomedical sensors are traveling together, being attached to the body of a patient. MCMAC categorizes the sensor nodes into a static network and a mobile cluster. The protocol defines a Reference Point Group Mobility (RPGM) model and a Random Waypoint Mobility (RWM) model to mimic the movement characteristics of mobile clusters and the individual node movement within a cluster. A frame in MCMAC is divided into an active and a sleep period. Since the slot assignment method is different for static and mobile nodes, the active period is further divided into static active slots (SAS) and mobile cluster slots (MCS). Static nodes communicate with each other in the SAS part by dynamically occupying a unique transmission slot in its two-hop neighborhood. The MCS part is used for nodes in a mobile cluster to communicate with each other. Since the size of a cluster can be small (a human body) and all enclosed nodes are typically within each other's one- or two-hop neighborhood, each slot in this part is assigned to exactly one node.

Despite a number of approaches exploiting the mobility of nodes for data collection in WSN, none address the issue of dynamic medium access control in a WSN with MDC (a scenario with all static source nodes and a mobile data collector).

4.2 PROPOSED SOLUTION

This section presents the Mobility Aware Sleep Wakeup Scheduling (MASWS) protocol which focuses on reducing energy consumption and improving data collection rates while supporting different mobility patterns of a mobile data collector. The core concept is based on adaptive time division multiple access (TDMA) where the sleep/wake-up duration is defined according to the mobility pattern of the MDC.

4.2.1 Assumptions

The proposal considers a wireless sensor network where several source nodes are deployed in a remote area for monitoring an environment of interest. The sensor nodes remain static once they are deployed. A mobile node is used to localize the static sensor nodes using a localization technique such as discussed in chapter 2. Once the localization of sensor nodes is completed, the mobile node can then be used as a mobile data collector (MDC). MDC can be either a mobile sink (which consumes the data itself) or mobile relay depending on the way it manages the collected data. Data collection is defined to be the flow of data from static source nodes to MDC. The MDC traverses the sensing field and periodically transmits beacon messages in order to announce its presence. Data transfer immediately follows after the discovery of MDC by static nodes. In this case, data collection takes place only during the contact time (when the static node and the MDC can reach each other). The goal is to get the most out of the contact time between source nodes and MDC, that is, to maximize the throughput, in terms of messages successfully transferred per contact while minimizing the energy consumption. While traversing the network area, MDC is assumed to know the transmission range and location of static sensor nodes (since the network is localized) as well as its own position using a GPS device.

4.2.2 System Model

The algorithm is specifically adapted for a cluster-based WSNs. In this scenario, the static sensor nodes are organized into non-overlapping clusters on the basis of the distance from each other. The sensor nodes sense their environment and save data in their buffers. Each sensor

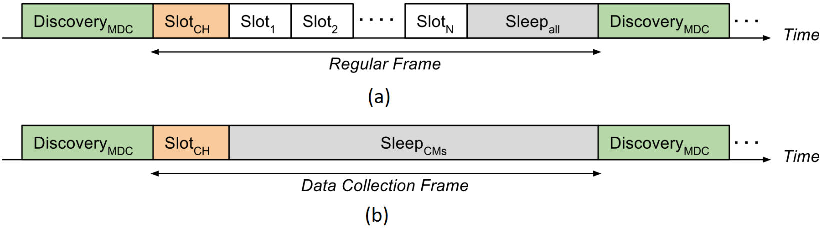


Figure 35 – Mobility Aware Sleep Wakeup Scheduling in Cluster based WSN

transmits data to its corresponding cluster head (CH), which in turn saves the data and eventually sends it to the MDC whenever it is in communication range. The cluster heads are responsible for coordinating the transmissions of the cluster members (CMs). CMs send data to their corresponding CH using single hop transmission while CH sends the collected data to the MDC using a single hop, thus minimizing collisions and message losses. In this case, the MDC should visit each CH individually and come within the closest possible distance to it.

Sensors nodes usually operate for a long time and send data only occasionally. The energy consumption of idle listening is equivalent to the energy consumption when sending or receiving, while much greater than the energy consumption when in sleep mode (AKYILDIZ et al., 2002). In this scheme, each CH acts as the centralized scheduler and controls data gathering from cluster members and then data dissemination to MDC. The CH sets up a Time Division Multiple Access (TDMA) schedule taking into account the sampling rate (packet generation rates) and transmission needs of each cluster member. TDMA protocols are more power efficient and avoid collisions since nodes in the network can enter inactive states until their allocated time slots. In addition, TDMA avoids overhearing and idle listening, which are typical problems in WSN medium access protocols.

In this scenario, data collection takes place only during the *sojourn time*, which is defined as the overall time spent by the MDC while traversing the *sojourn distance* (path of MDC inside the communication zone of a static node). It is assumed that the trajectory and speed of the MDC are not controllable. In addition, it is considered that the arrival time of the MDC into the coverage zone of a static node can be either random, predictable, or controlled. Similarly, the *inter-collection-time* of the MDC is defined as the actual period of time elapsed from the beginning of a data collection to the beginning of the next one (time between data collection schedules).

The CH generates a TDMA schedule (see Figure 35 (a)), referred to as a *Regular Frame* and shares it with its cluster members. One slot ($Slot_{CH}$) in the schedule is reserved for transmissions from the CH to the cluster members which is also used for synchronization purpose. All nodes remain active in $Slot_{CH}$. Similarly, one slot ($Slot_n$) is reserved for each cluster member to send its data to CH. A Regular Frame consists of an T_{Active} portion (all active slots) and an inactive period $Sleep_{all}$, where all nodes remain in sleep mode to save power. The sampling rate determines the data generation capability and transmission needs of each node. As a fair policy, nodes with higher sampling rate must have more amount of time allocated in TDMA schedule than nodes with lower sampling rate. This ensures that each node gets enough time to send its generated data to their corresponding CH and no time slot is wasted. It is assumed that the CH knows the sampling rate (or transmission needs) of each of its member node from their JOIN message and use the following equation to generate TDMA time slots for each of them.

$$slot_i\,(unit\,time) = \frac{(T_{Active})SR_i}{SR_T}(unit\,time)$$

Where, SR_i is the sampling rate of node i, SR_T is the sum of sampling rates of all member nodes and T_{Active} is the active portion (sum of all active slots) in a frame.

Each source node generates data and sends it to the CH in its scheduled slot. The packets from each cluster member node are tagged with a sequence number (SQ1). The CH node sends in its TDMA slot $Slot_{CH}$ the SQ1 of the last packet received successfully from each cluster member. In this way, each source node can keep track of the packets that were received successfully by the CH.

Since a CH has to transfer data to the MDC if it is nearby, a slot $Discovery_{mdc}$ is reserved after each TDMA frame to find whether the MDC is in communication range or not. The CH aggregates sensors data from all cluster members. Meanwhile, the MDC periodically sends beacons messages in order to advertise its presence. When a CH knows that the MDC has entered into its coverage zone by hearing a beacon message, it starts transferring the aggregated data to MDC. These data messages are also tagged with a sequence number (SQ2). The MDC sends in the beacon message the SQ2 of the last message from that cluster that was received by MDC. Thus, each CH knows which messages were received successfully at the MDC and can retransmit messages if necessary.

The actual *sojourn distance* or *sojourn time* of a mobile node with a static node can be calculated well in advance if the path of the mobile node and location and communication range of the corresponding static nodes are known, as discussed in the previous chapter. MDC calculates its sojourn time based on its current speed and trajectory (using the method described in chapter 2) and announces this information in the beacon message. Upon receiving the beacon message, the CH instructs its cluster members to go to sleep mode for at least the corresponding sojourn time, just announced by MDC. In other words, it generates a new TDMA frame, called *Data Collection Frame*, as shown in Figure 35 (b). Here the length of the $Sleep_{CMs}$ slot is at least equal to the corresponding sojourn time of MDC. All cluster members go to sleep mode in this slot. As a result, during the data transfer process, the MDC is only in communication with one node (the corresponding CH). This not only reduces energy consumption but also helps avoiding collisions and interference between sensors-to-CH communication and CH-to-MDC communication.

From the cluster head point of view, the overall data collection process can be split into four main phases, as shown in the state diagram depicted in Figure 36. A CH performs a discovery phase in order to detect the presence of MDC. The discovery phase is performed periodically or at some specific times, depending on the mobility pattern of MDC. Once a CH hears a beacon message from an MDC, it sends a reply to MDC, generates a sync and scheduling message for cluster members (using the data collection TDMA frame in Figure 35-b), switches from the *discovery* state to the *data transfer* state, and starts transmitting data to the MDC. If it does not hear any beacon message from MDC in the specified time, it generates a sync and scheduling message for cluster members (using the regular TDMA frame in Figure 35-a) and switches from the *discovery* state to the *data gathering* state. The scheduling message is different in each case. When switching to *data transfer* state, it instructs all cluster members to go to sleep mode for a specified period of time, while switching to *data gathering* state, it defines communication slots and sleep duration for its cluster members. The length and existence of discovery phase depend on the application scenario and mobility pattern of MDC. A CH may generate a TDMA schedule and switch directly from the *sleep mode* state to *data gathering* state, if it does not need to detect an MDC (if its arrival is not expected in the case of predictable mobility or not needed in case of controlled mobility). CH switches to sleep mode once data transfer to

MDC is finished and the frame is not finished yet or when it reaches its scheduled sleep mode time. A CH in the data transfer state remains always active to exploit the contact as much as possible. On the other hand, the MDC enters the data collecting phase as soon as it receives the first reply from the cluster head, and stops transmitting beacon messages.

Selective repeat based communication protocol (KUROSE, 2005) is adopted during the data transfer phase from CH to MDC. Selective repeat communication is based on a window-based ARQ protocol with selective retransmission for a particular window size (assumed to be equal to W messages). It should be noted here that the ACK messages are used for implementing a retransmission mechanism as well as an indication of the MDC presence in the communication area. The data collection phase stops either when the CH has no more packets to transfer or the MDC is not reachable anymore. When a CH misses N_{ack} consecutive acknowledgments, it confirms that the MDC is out of range and stops sending data. Similarly, the MDC assumes that the data collection is over when it does not receive any packets in a given period of time.

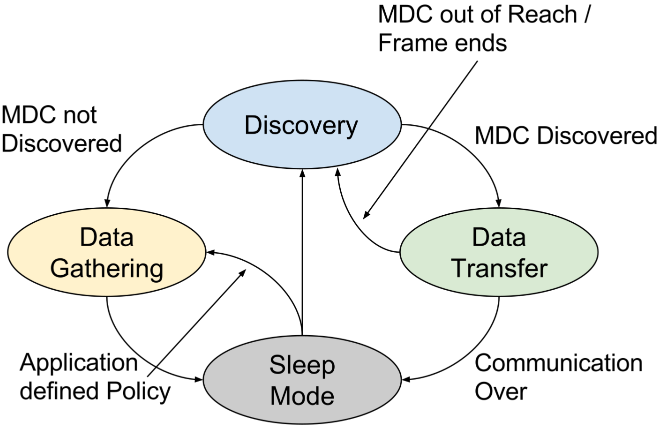


Figure 36 – State Diagram for Cluster Head

Figure 37 shows the three state diagram of static source nodes. A static source node remains in sleep mode and wakes up either in the beginning of its slot to start transferring data to the CH or at the end of the frame to receive the next scheduling message from CH.

If MDC's beacon messages are spaced by a beacon period of T_B

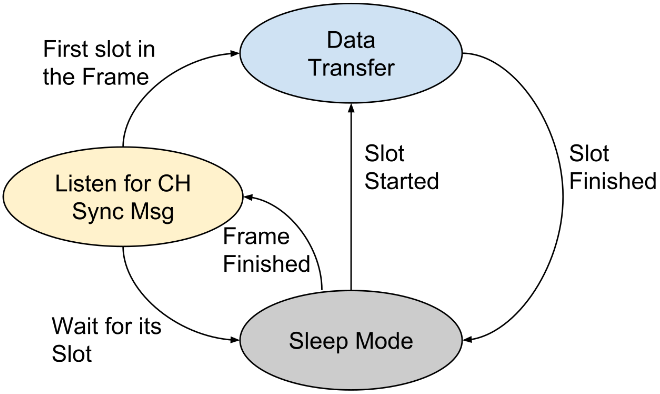


Figure 37 – State Diagram for Static Nodes

then the length of discovery phase should be $Discovery_{mdc} \geq T_B$ so that at least one complete beacon can be received, provided that the MDC is in the contact area.

Data transfer between MDC and CH is possible only during sojourn time, so this time should be fully exploited and wisely used. Hence, it is important for the MDC to enter the coverage zone of a CH when it is in discovery mode (or it is important for a CH to be in discovery mode when the MDC is entering its coverage zone) so that the CH could timely detect it and start transferring data. If these two events, the time of entrance of the MDC in the communication range of the CH (named t_{enrty}) and the duration of the discovery phase of a CH (named $T_{discovery}$, meaning the size of the $Discovery_{mdc}$ slot), are not synchronized (not overlapped nor happen at the same time) then MDC will have to wait till the start of next $Discovery_{mdc}$ slot so that it can be discovered and data transfer is initiated. In this case, it might happen that the TDMA frame is too long and the valuable contact time is wasted in waiting. That is why special efforts must be made so that the MDC could enter the coverage zone of a CH in $Discovery_{mdc}$ slot or if this could not be guaranteed, to get the most out of the sojourn time of the MDC with each CH (a scarce resource in this scenario).

Different mobility patterns of MDC require proper adjustments in the design of MASWS proposal, described as follows.

4.2.2.1 MDC with Random Mobility

Random mobility of MDC is characterized by its irregular arrivals which take place randomly with some probability distribution. Obviously, when the mobility pattern is completely random, it is hard to predict the arrival time of MDC. In this case, the static nodes (CH) should continuously perform MDC detection. The CH node does this in $Discovery_{mdc}$ slot at the end of every TDMA frame. However, it could still not be guaranteed that the MDC enters the communication zone of CH, exactly in $Discovery_{mdc}$ slot. Hence, the goal in this scenario is to maximize the usage of available sojourn time. The best strategy, in case of completely random mobility, is to keep the size of $Regular\,Frame$ much less than the average sojourn time of MDC, so that if the entrance of MDC and $Discovery_{mdc}$ slot are not aligned, a very small portion of the sojourn time is wasted and the rest of the contact is fully utilized. For instance, if the communication radius of a CH is 50m and MDC is moving with 2m/s then the maximum sojourn time of MDC with the corresponding CH is 50s and the average sojourn time is 25s since it is not known to the CH how long the sojourn distance of the MDC is (MDC can pass exactly over CH or far away at the communication border). So a simple rule of thumb, if the average speed of MDC is available to CH, is to keep $Regular\,Frame$ much less than 25s. i.e.

$$Regular\,Frame \ll \frac{r_{CH}}{v_{MDC}}$$

In this way, in the worst case (if t_{enrty} do not occur in $T_{discovery}$), only a time equal to $(Regular\,Frame + Discovery_{mdc} + Slot_{CH})$ will be wasted and the rest of the contact will be fruitfully utilized by collecting data from CH.

Similarly, another less suitable solution is to keep the length of $Regular\,Frame$ intact and increase the length of $Discovery_{mdc}$ slot. This will increase the probability of MDC entrance in the communication zone of CH during $Discovery_{mdc}$ slot. However, there will still be the chance that the two events (t_{enrty} and $T_{discovery}$) do not overlap. The damage, in this case, will be much more since the length of $Regular\,Frame$ is not minimized and $Discovery_{mdc}$ slot is extended and it is possible that a substantial size of contact time will be wasted depending on the size of $Discovery_{mdc}$ slot. It must be noted here that the CH starts transmitting data to MDC once $Discovery_{mdc}$ slot is finished and cluster members are instructed to go to sleep mode in $Slot_{CH}$. Hence, an extension of $Discovery_{mdc}$ slot gives a trade-off between improving the probability of MDC being on time or risk of losing a greater portion of sojourn time.

4.2.2.2 MDC with Predictable Mobility

The predictable mobility of MDC is characterized by regularity in the arrival of the MDC which enters the communication range of a CH at specific and usually periodic times. Here CH has some or exact information about MDC arrival and contact time. In this case, it is possible for the CH to exploit some knowledge on the mobility of MDC, to avoid unnecessary detection and to further reduce its energy consumption (JUN; AMMAR; ZEGURA, 2005). In the case of predictable mobility, MDC may follow either a strict and accurate schedule or a loose and probabilistic schedule.

If the MDC follows and maintains a very accurate and exact schedule then the CH can know an exact time when the MDC will enter its coverage area, and can thus generate TDMA Frames suitable for MDC arrival. The goal is to keep the CH in the discovery mode exactly when the MDC is about to enter the coverage zone of the CH. For instance, if the MDC is expected to enter the coverage zone of a CH exactly at t_{entry} then the last TDMA regular frame is adjusted (reduced or enlarged) so that the regular frame ends exactly at t_{entry}. Obviously, such approach requires that the mobility of the MDC is accurate, strict and controlled enough to obey on-time arrival. In this case, there is no need to detect the presence of MDC after each TDMA frame and hence the duration of $Discovery_{mdc}$ slot could be changed to zero at other times while it should be greater than the MDC beacon interval when MDC is approaching at t_{entry}.

If the MDC follows a probabilistic schedule then the CH can know the exact interval (not a specific time) when the MDC is expected to enter its coverage area. For instance, the CH knows with 100% probability that the MDC will enter its coverage zone between t_1 and t_2. In this case, CH can follow a normal TDMA structure for $t < t_1$ and $t > t_2$. For $t_1 < t < t_2$, CH should follow the same rule specified for random mobility of MDC, such as to keep $Regular\,Frame$ much less than the average sojourn time. In this case, $Discovery_{mdc}$ slot could be reduced to zero at other times while it should be greater than beacon interval of MDC when $t_1 < t < t_2$.

Here the efficiency of the TDMA scheme is further improved by exploiting knowledge on the mobility pattern of the MDC, such that the CH tries to detect MDC only when it is likely to be in contact. If contact times are known with a certain probability, CHs can be awake only when they expect MDC to be in their transmission range. In both strictly as well as loosely scheduled MDC, the duration of $Discovery_{mdc}$ slot is kept zero for $t \neq t_{entry}$ and $t < t_1$ and $t > t_2$, since the MDC is not expected to arrive in these intervals and there is no need to reserve a slot to detect MDC.

Initially, static nodes start with no prior knowledge on the mobility pattern of MDC. Eventually, they can learn by observing the arrivals of the MDC in each round of data collection. In addition, MDC could also inform each CH about its next visit if the next visit's schedule is decided and this information is available in advance.

4.2.2.3 MDC with Controlled Mobility

Controlled mobility is characterized by the fact that the MDC actively and deliberately changes its location by controlling its trajectory and speed (FRANCESCO; DAS; ANASTASI, 2011). This gives MDC an additional power which can be effectively exploited in designing data collection protocols. It should be noted that controlled mobility simplify the problem of discovery by visiting each node at a specific time. MDC with controlled mobility results in two possibilities. In the first case, MDC is controllable and CHs knows its exact arrival time. In the second case, MDC is controllable but CHs have no information regarding its arrival time.

The first case is exactly similar with an MDC which has an accurate and exact schedule as discussed in section 4.2.2.2. Here CHs make the necessary adjustments to the TDMA structure (reduced or enlarged) so that the regular frame ends exactly at t_{entry} (arrival time of MDC). In this case, there is no need to detect the presence of MDC after each TDMA frame and hence $Discovery_{mdc}$ slot could be changed to zero at other times while it should be greater than beacon interval when MDC is approaching. In the second case, CH has no clue regarding the arrival of MDC however MDC knows TDMA scheme of the CH. In this case, each CH maintains and continues its normal TDMA scheme. Here MDC do the right job by controlling its motion and enters the communication zone of each CH at the right time when they are in discovery mode. It must be noted here that in this case, normal TDMA structure of the network should be known to MDC at the design time.

If the CH has some information about the arrival of MDC, it can use it to avoid unnecessary listening otherwise if completely blind it has to switch to discovery mode after each TDMA frame. In addition, there is no need to keep the length of regular TDMA frames much less than the average sojourn time as in the case of random mobility thanks to the controllability feature of MDC. In this situation, the MDC can either schedule its trip properly or actively control its motion (slowing down or speeding up as discussed in the previous chapter, in order to reach each CH on time.

In both predictable and random mobility patterns, MDC arrival can be either stationary or dynamic. In the first case, MDC arrivals are usually periodic (repeated in a given time) while in the second case, contacts show some degree of periodicity, but their period or pattern can change from time to time. For stationary mobility patterns, nodes can learn the schedule of MDC only once in the beginning, since the arrival pattern does not change with time. However, dynamic mobility requires continuous monitoring, so that the cluster heads can adapt to changing mobility patterns of MDC.

4.3 SIMULATION AND RESULTS

4.3.1 Simulation Setup

To evaluate the performance of MASWS, simulated environments were implemented in Omnet++ tool. The simulated scenario corresponds to random deployments of sensor nodes over a 200 x 200 m2 area for periodic reporting of sensed data. In such applications, the data of interest (e.g., temperature, vibrations) are sensed and reported periodically to MDC. In all simulated experiments, IEEE 802.15.4 radio model with 2.4 GHz physical layer was used. The propagation model used in the simulation was the *free space*. The transmission range of each CH is approximately a circle with a 50m radius.

A simulation scenario was developed where 6 to 60 nodes are randomly deployed and organized into 3 clusters on the basis of the distance from each other. The length of the track and hence each data collection lap is 550m approx. The MDC path is assumed to be a straight line while passing through each cluster zone. As soon as the mission begins, the MDC starts traversing the network area on the pre-defined track and starts collecting data by periodically sending a beacon message. Any leftover data with a CH is collected in the next lap. For simplicity, it is assumed that MDC has 100 m sojourn distance with each of the three CH. The CH nodes are assumed to collect and keep data till one complete inter-collection-time.

It is assumed that a sensor reading takes 8 bytes of storage in the memory. 50% of nodes in a cluster take 18 readings per minutes while remaining 50% take 6 readings. This generates an average of 1840 packets of data per node in 24 hours where each packet has 75 bytes. It is considered that the MDC is used after each 24 hours (inter-collection-time) to collect all these samples for further processing.

The mobility of MDC is considered to be predictable where the MDC arrivals are periodic and the inter-collection-time is fixed to be $24hours \pm 30min$ (expected MDC interval to be 60 minutes). This mobility pattern represents the case where the arrivals of MDC is expected in an interval.

The adopted parameters settings for the simulations are shown in Table 5. The radio energy consumption during the idle (monitoring channel) state and receive state is assumed to be the same. Simulation parameters used for evaluating energy consumption are chosen according to the methodology used in (FRANCESCO et al., 2010).

4.3.1.1 Performance Metrics and Parameters

The following performance metrics are considered in the evaluation of the proposed protocol.

- **Packets Collected in the First Round:** Total number of packets successfully collected by the MDC during the first complete round (lap) on its specified trajectory.
- **Number of MDC Rounds Required:** Total number of MDC rounds (laps) required for successfully collecting all data generated during one *inter-collection-time.*
- **Total MDC Traverse Time:** Total time taken by the MDC for successfully collecting all data generated during one *inter-collection-time.*
- **Energy Consumption per Packet:** Amount of energy consumed per successfully collected packet by the MDC after completing the data collection process. It must be noted that this does not include the energy spent by MDC during data collection; only the energy of static nodes is considered. A rough estimation of the energy spent by MDC can be made from the *Total MDC*

Traverse Time which is directly related to the total energy spent by MDC.

Table 5 – Parameters adopted for MASWS Simulation

Parameters	**Values**
Nodes per Cluster	2, 4, 6, 8, 10(default), 12, 14, 16, 18, 20
Regular Frame Size (s)	1, 2(default), 3, 4, 5, 6, 7, 8, 9, 10
MDC Speed (m/s)	1, 2, 3(default), 5, 8, 10, 12, 14, 16, 18, 20
$Discovery_{mdc}$ (s)	0.1, 0.2(default), 0.4, 0.6, 0.8, 1, 1.5, 2
Beacon Period (T_B) (ms)	50
Active Time (T_{Active})(s)	0.5, 1(default), 1.5, 2, 3.5, 4, 4.5, 5
Inactive Time ($Sleep_{all}$)(s)	0.5, 1(default), 1.5, 2, 3.5, 4, 4.5, 5
ARQ Window Size (W)	20
N_{ack}	5
Radio Transmit Power	49.5mW
Radio Receive Power	28.8mW
Radio Idle Power	28.8mW
Sleep Mode Power	0.6μW

4.3.2 Simulation Results

The performance of MASWS is compared with 802.15.4 beacon enabled mode (for SO=0; BO=1 and SO=0; BO=2) in terms of the performance metrics, discussed in 4.3.1.1.

MASWS performs better in terms of data collection rate, as stated in Figure 38. It shows the number of packets successfully collected by the MDC in the first round of the data collection process. As the number of nodes per cluster increases, so does the amount of data transfer increase from CH. With less number of nodes per cluster, MASWS and 802.15.4 beacon-enabled mode behave similarly. This is due to the fact that each CH wishes to transfer less number of packets and that the MDC has enough sojourn time to collect it completely. When there are more nodes per cluster and hence more data with the CH, MASWS is able to grab more data during one round of data collection. This is due to the fact that in MASWS, all cluster members are in sleep mode during the data transfer to MDC. All sojourn time is completely dedicated to data collection with no collisions at all. In

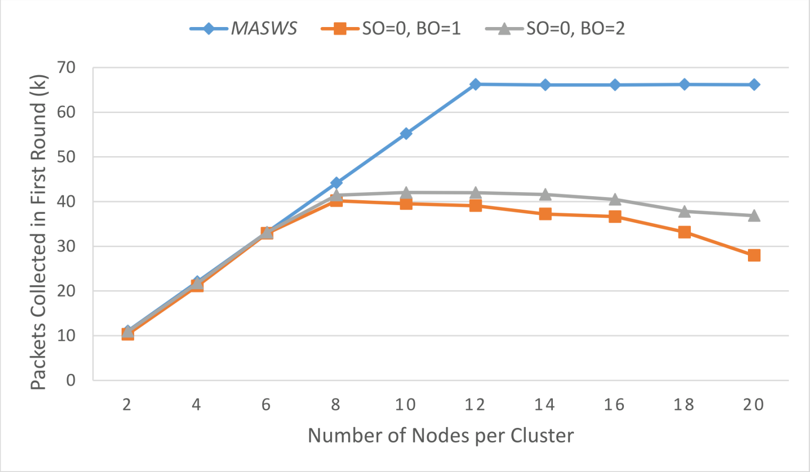


Figure 38 – Packets successfully collected by MDC with different number of nodes per cluster

802.15.4 beacon enabled modes the valuable sojourn time of the MDC is not completely dedicated to data collection from CH.

Figure 39 presents again the number of packets successfully collected by MDC in the first round of data collection process. However, in this case, the data generation capability of all nodes is doubled than the default value. This figure shows that the advantage of MASWS is clearly visible when CH has enough data to transfer to MDC. As the number of nodes (or data generation capability) increases so does the data collection rate. But once a saturation point is reached, it can be seen that MASWS clearly outperform in data collection rates.

Figure 40 shows the number of packets successfully collected by MDC in the first data collection round. It can be seen that as the size of the regular frame is increased, the data collection rates declines. As mentioned earlier, the size of the regular frame must be kept much less than the average sojourn time. The more the size of the regular frame, the more the sojourn time will be wasted in waiting till the next discovery phase (if the MDC Entry and Discovery phases are not aligned). Similarly, wasting more sojourn time means less number of packets to be collected in a single round of MDC.

Data collections per round also drop (see Figure 41) with increasing the MDC speed. Higher MDC speeds allow less sojourn time in a

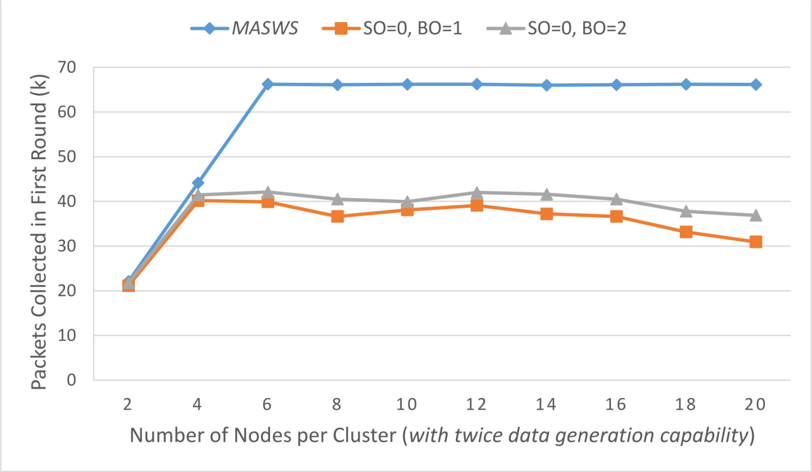


Figure 39 – Packets successfully collected by MDC with different number of nodes per cluster, but with twice data generation capability

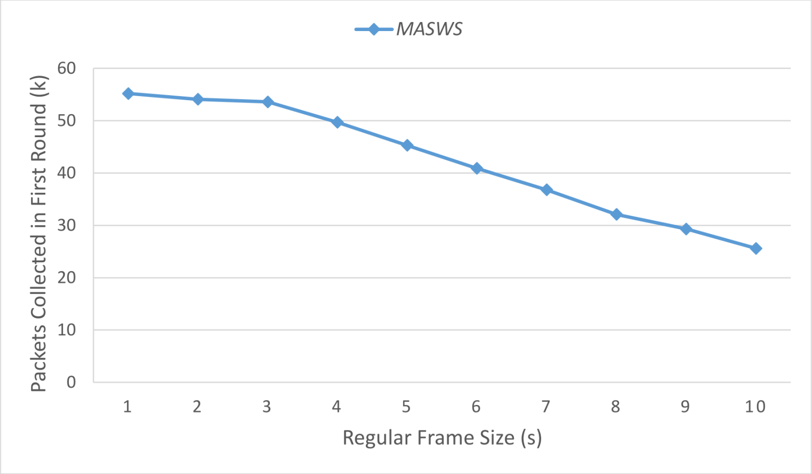


Figure 40 – Packets successfully collected by MDC with different frame size

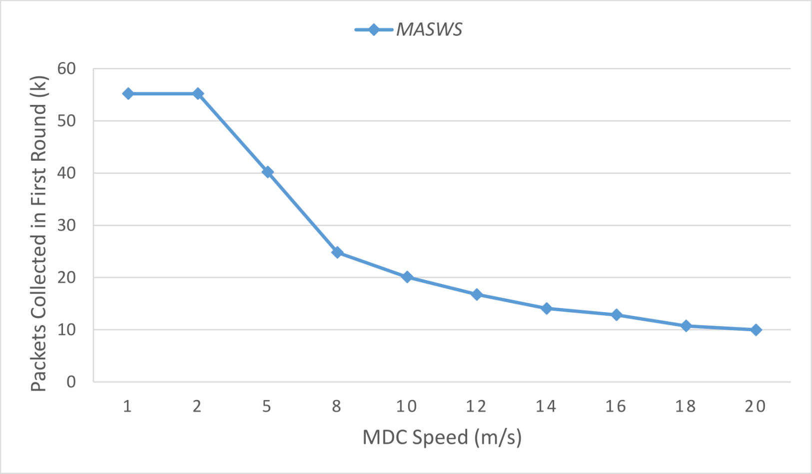


Figure 41 – Packets successfully collected by MDC with different MDC speeds

single data collection lap thereby resulting in fewer packets collected.

The effect of increasing the value of $Discovery_{MD}$ has a very slight effect on the number of packets successfully collected by MDC in one round as shown in Figure 42. The slight decrease in data collection rate with the increase in discovery phase may due to the fact that with large $Discovery_{MD}$, the MDC has to wait a little more for a discovery phase if the MDC Entry and Discovery phases are not aligned.

Figure 43 shows the data collection rate with different expected MDC interval time. It can be seen that as this interval goes on increasing, data collection slightly drops. In the case of larger interval, the probability of the alignment of MDC entry and Discovery phase decrease, which increases the chance of wasting some portion of sojourn time in alignment.

Besides, increasing the interval of expected MDC arrival results in increasing the energy consumption per successful packet collected by MDC, as shown in Figure 44. A larger value for an interval of expected MDC arrival means more frequent discovery phases and hence more idle listening. It should be noted here that the value of 24 hours for expected MDC arrival means complete random arrivals of MDC. In this case, CH continuously performs discovery phase. The value of 0.0003 hours (which corresponds to 1 sec) for expected MDC arrival means

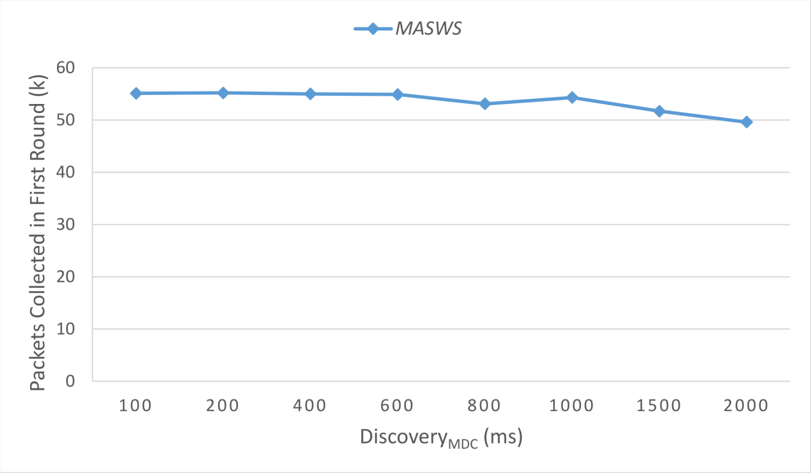


Figure 42 – Packets successfully collected by MDC with different value for $Discovery_{MD}$

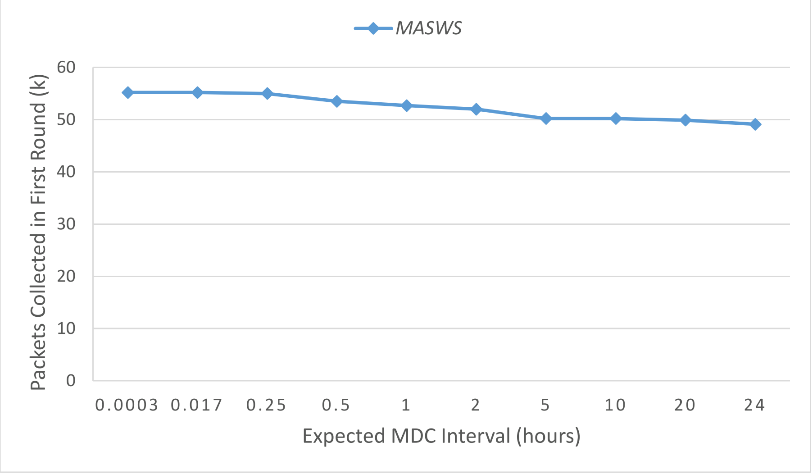


Figure 43 – Packets successfully collected by MDC with different expected MDC interval

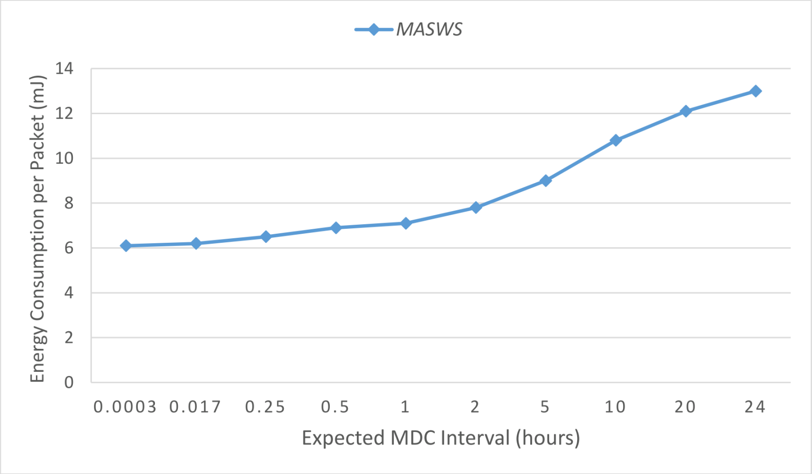


Figure 44 – Energy consumption per packet with different Expected MDC interval

controlled arrival, where the MDC arrives exactly at a specific time. In this case, the MDC performs the discovery phase only once, just before the expected time. The values of expected MDC arrival in the middle represents predictable mobility (predictable arrivals) of MDC, where the MDC is expected to arrive with 100% probability in between a given interval.

The overall maximum data collection rate achieved during the simulation experiments for MASWS and 802.15.4 beacon enable modes are presented in Figure 45. Clearly, MASWS outperforms 802.15.4 beacon enabled modes due to collision avoidance and a more suitable sleep wakeup schedule.

Increasing the number of nodes in the network increase the data generation capability of the network. Hence, the total number of MDC rounds required to collect all the data generated in the inter-collection-time also increases, as shown in Figure 46. This is due to the fact that the sojourn time of MDC remains the same and that's why it requires more laps for grabbing more data. However, due to better data collection rate, MASWS requires less number of MDC rounds than 802.15.4 beacon enabled modes.

It is obvious that the total traverse time of MDC is directly related to the number of MDC rounds. Hence, the total traverse time of MDC also increases with increasing the number of MDC rounds as

Figure 45 – Maximum data collection rate

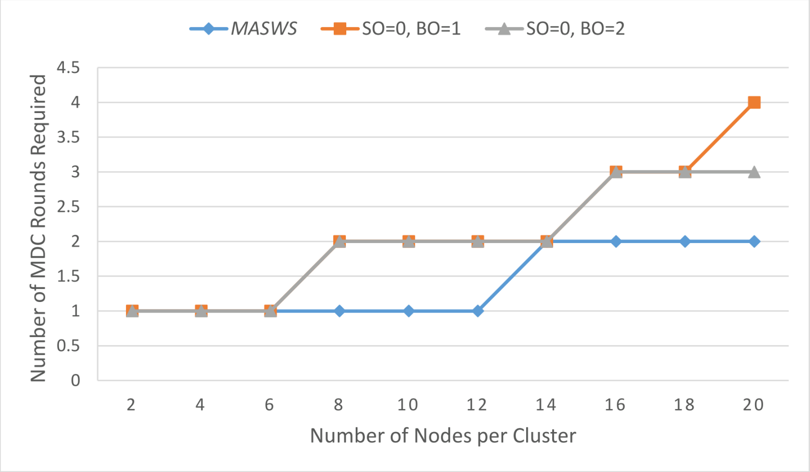


Figure 46 – Total MDC rounds required with different number of nodes per cluster

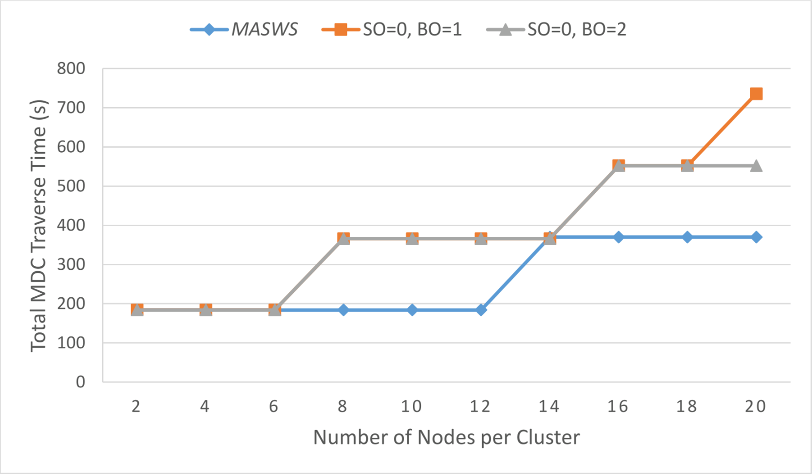


Figure 47 – Total MDC traverse time required with different number of nodes per cluster

shown in Figure 47. It must be noted here that the energy / fuel consumption of MDC during traversing the network was not directly measured from the simulated experiments. However, in the case of MDC with uniform speed, it can be easily deduced that the energy consumption of MDC is directly proportional to the traverse time of MDC.

Increasing the regular frame size causes a reduction in the data collection rates, as it increases the number of MDC rounds as shown in Figure 48.

The total number of MDC rounds also increases rapidly with increasing the MDC speed. Greater MDC speeds considerably decrease the sojourn time with the corresponding CH, which results in increasing the number of laps required to collect all data produced in the inter-collection-time, as shown in Figure 49.

MASWS also performs better than 802.15.4 beacon enabled modes in terms of energy consumption per successful packet collected by MDC, as shown in Figure 50. As emphasized earlier, MASWS completely avoid collisions, overhearing, and idle listing by properly adapting the sleep/wake-up schedule, thereby minimizing the energy consumption of nodes.

Increasing frame size decreases data collection rates, which also

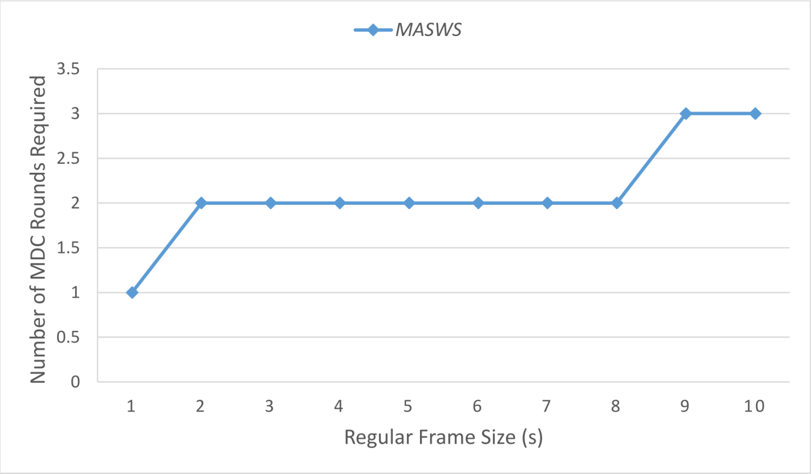


Figure 48 – Total MDC rounds required with different frame size

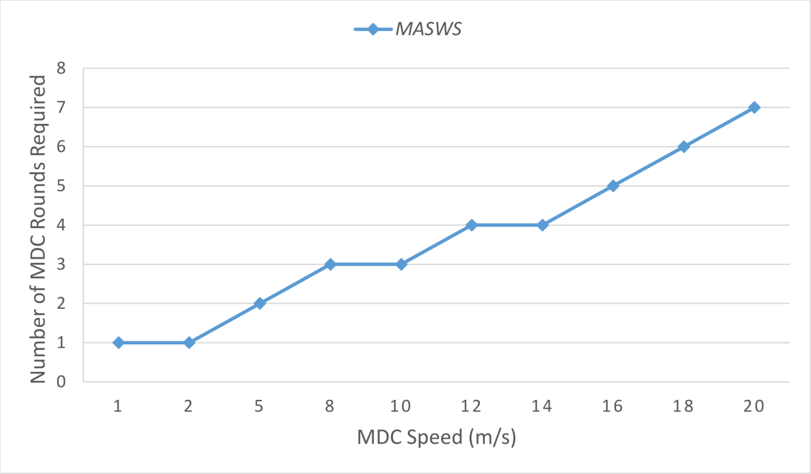


Figure 49 – Total MDC rounds required with different MDC speeds

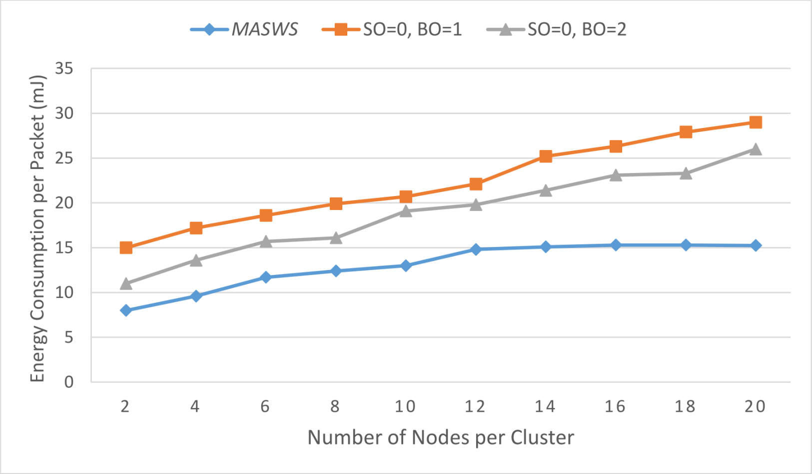


Figure 50 – Energy consumption per packet with different number of nodes per cluster

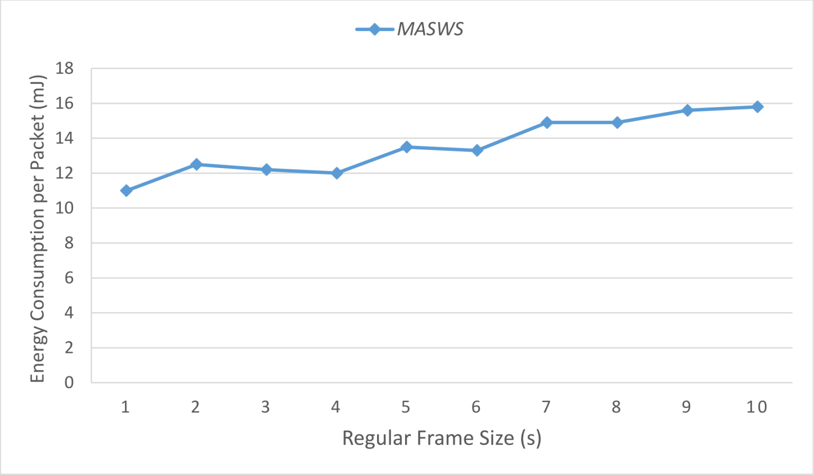


Figure 51 – Energy consumption per packet with different frame size

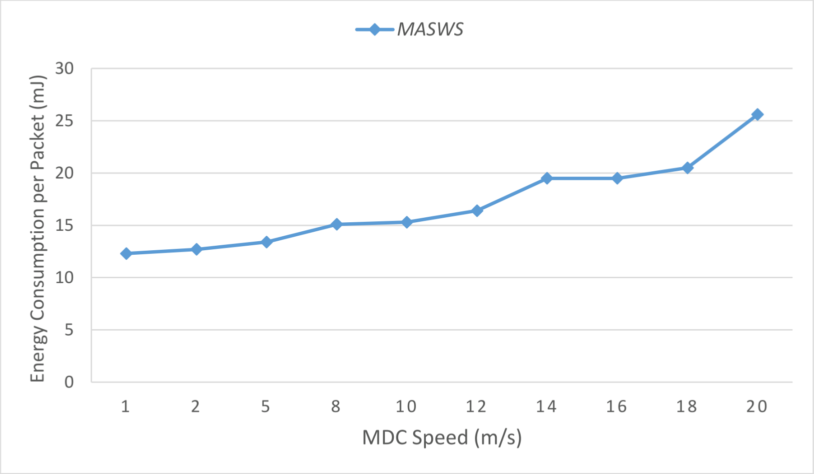


Figure 52 – Energy consumption per packet with different MDC speeds

results in increasing energy consumption per successful packet collected by MDC as shown in Figure 51. With larger frame size, the probability of alignment of MDC entry and discovery phase decreases and hence results in a higher waste of sojourn time and energy.

Increasing MDC speed decreases data collection rates, which also results in increasing energy consumption per successful packet collected by the MDC, as shown in Figure 52.

Increasing the value of $Discovery_{MDC}$ decreases data collection rates, which also results in increasing energy consumption per successful packet collected by MDC as shown in Figure 53. The Greater value of $Discovery_{MDC}$ mean more idle listening and hence increase in energy consumption.

4.4 CHAPTER CONCLUSIONS

This chapter discussed several benefits of implementing a mobility aware sleep wakeup scheduling in a data collection scenario of WSN with MDC. The design goal focused on minimizing energy consumption and improving data collection rates while supporting different mobility patterns of a mobile data collector. An adaptive time division multiple access scheduling is described where the sleep/wake-up duration is defined according to the mobility pattern of MDC. First. the CH sets

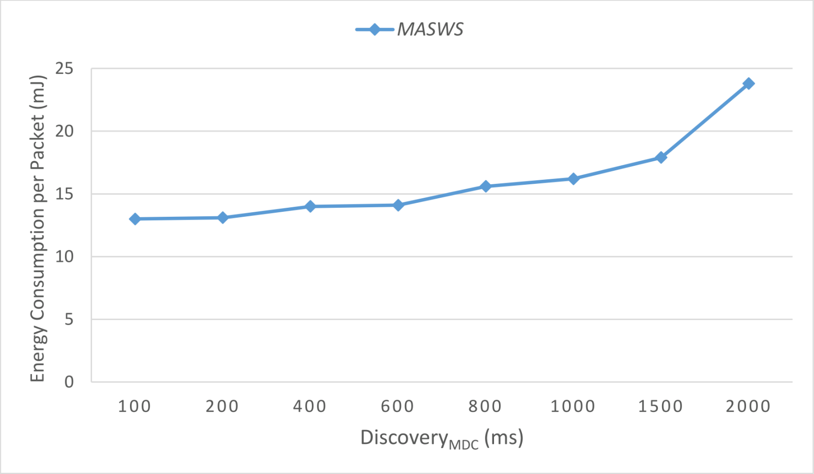


Figure 53 – Energy consumption per packet with different value for $Discovery_{MDC}$

up a TDMA scheme taking into account the transmission needs of each cluster members, thereby avoiding the packets collisions and idle listening. When the MDC approaches, the CH instruct cluster members to go to sleep mode just before initiating the data transfer to MDC. As a result, the MDC is only in communication with the corresponding CH. This not only reduces energy consumption but also helps avoiding collisions and interference between sensors-to-CH communication and CH-to-MDC communication. The proposed solution performed better during the simulated experiments in terms of energy consumption and data collection rates, which means it also saves significant MDC traverse time.

5 CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

This thesis explored techniques related to exploiting and optimizing controlled mobility in wireless sensor networks for performance and efficiency gains. The focus was on a WSN scenario that consists of several static source nodes with one or more mobile nodes. The mobile nodes are assumed to be not resource constrained and are used to assist in different operation of the network, such as localization and data collection.

In the first case, in chapter 2, the localization of static sensor nodes using a controlled mobile element was discussed. Position information of static nodes is very important in WSN. It helps in effective coverage, routing, data collection, target tracking, and event detection. The proposed localization scheme included a technique to estimate the locations of static source nodes with the help of a controlled mobile node and simple geometric techniques. First, the mobile node scans the network area while broadcasting periodic beacon messages and find at least three boundary points on the communication circle of the source nodes. Then, based on the estimated coordinates of the boundary points, together with elementary geometry and algebra, each sensor node calculates its location coordinates. The scheme does not require extra hardware or data communication and does not require the static sensor nodes to spend energy on any interaction with their neighboring nodes. The proposed algorithm is scalable, distributed, and power efficient since the mobile node only broadcasts beacon messages. The scheme showed good level of accuracy in the presence of obstacles. Obtained simulation results showed that the localization accuracy can be kept low and well adjusted by properly selecting the speed, beacon interval, and scan pattern of the mobile node.

Chapters 3 and 4 considered a cluster-based WSN, where the static sensor nodes are organized into non-overlapping clusters. The cluster heads are responsible for coordinating the transmissions of the cluster members. Sensors transmit data to their corresponding CH, which in turn saves the data and eventually sends it to the MDC whenever it is in communication range.

More specifically, chapter 3 focused on investigating the viability and usefulness of a dynamic speed control for a mobile data collector in a cluster-based WSN. In this case, the elements affecting the data collection process using an MDC was discussed. Also, an adaptive algorithm, along with its control parameters, was proposed so that the MDC can autonomously control its motion while collecting data in the network. The discussed parameters allow the speed of the MDC to be adjusted at run time in order to adaptively improve the data collection process. Built-in intelligence helps the system adapting to the changing requirements of the data collection process. The proposed scheme showed significant advantages for sparsely deployed, large scale sensor networks, and heterogeneous networks (where sensors have variable sampling rates). The simulation results showed a significant increase in the data collection rate and reduction in the overall traverse time and number of laps that the MDC spends for data gathering.

Finally, chapter 4 discussed the benefits of implementing a mobility aware sleep wakeup scheduling in a data collection scenario of cluster-based WSN with an MDC. The design goal focused on minimizing energy consumption and improving data collection rates while supporting different mobility patterns of mobile data collector. The core concept was based on an adaptive time division multiple access scheduling where the sleep/wake-up duration is defined according to the mobility pattern of MDC. The protocol was discussed for an MDC with random, predictable, and controlled mobility in a cluster-based WSNs. The CH sets up a TDMA scheme taking into account the transmission needs of each cluster members, thereby avoiding the packets collisions and idle listening. When the MDC approaches, the CH instruct cluster members to go to sleep mode just before initiating the data transfer to MDC. As a result, the MDC is only in communication with the corresponding CH. This not only reduces energy consumption but also helps avoiding collisions and interference between sensors-to-CH communication and CH-to-MDC communication. The proposed solution performed better during the simulated experiments in terms of energy consumption and data collection rates and saved significant traverse time of MDC.

5.2 FUTURE WORK

Several improvements and extensions to the works proposed in this thesis can be made and is intended to be carried out in future work, in a intended post-doc position.

In this regard, an extension to the localization scheme proposed in this thesis is in pipeline. The extended localization scheme should focus on the same technique of using a controlled mobile node but with directional antennas. Directional antennas result in more focussed transmissions with reduced energy consumption. The current scheme of EGL requires three boundary points on each sensor node, which requires at least two passes of the mobile node through the coverage zone of each node. Here, the third boundary point merely helps in choosing the right location obtained from the first two boundary points. The extended version will focus on making use of only two boundary points with extra information from directional antennas without compromising the accuracy of position information. This will not only reduce the traverse time of the mobile node but also the localization time as well as the energy consumption. The detection and correction techniques for errors in the boundary points is also intended to be improved.

Similarly, the adaptive speed approach proposed for modifying the MDC speed is capable of adapting to changes in data collection rates. Topology modifications, however, are still not tolerated in this proposal and should be subject of future investigation. Similarly, adaptive speed control along with an adaptive trajectory control (not considered in this work) would be an interesting topic to investigate. Adaptive speed control in a scenario with non-clustered WSN is also under consideration. In such case, static sensor nodes are deployed in the sensing area without any cluster formation. From non-cluster WSN it is meant a network where the MDC can be simultaneously in range of two or more than two sensor nodes as shown in Figure 54. The sensor nodes measure their environment, save data in their buffers, and eventually transfer it to the MDC whenever it is in communication range. The nodes do not cooperate with each other and only transfer their data to the MDC. In case of non-clustered WSN, the MDC should accommodate data transmission from all surrounding nodes and hence should adjust its speed based on the data and distance with all engaged nodes. Besides, if the MDC does not have location information of the source nodes (if the network is not localized). It would be interesting to investigate how the MDC would find sojourn distance in this case (since sojourn distance is an important factor in deciding and calculating an optimum speed).

It is also planned to adopt and investigate the sleep/wake-up scheduling for a non-clustered WSN. In this case the static source node should coordinate in a manner so that the data transfered to the MDC is maximized and energy consumption is minimized. Similarly, in dense and non-clustered networks, the data generated by each node may be important and hence each node may desire an equal opportunity and

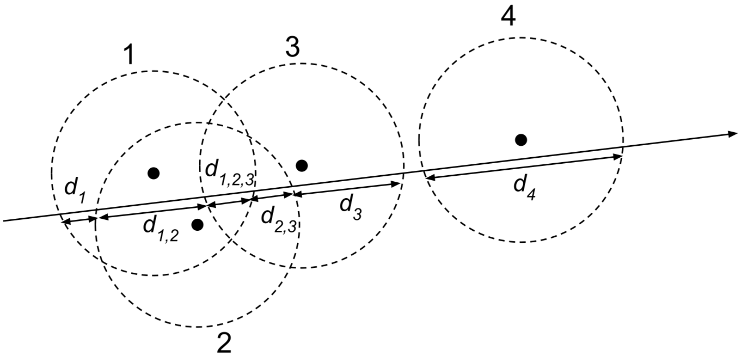


Figure 54 – MDC path in a non-clustered WSN

time to access the medium, i.e., transferring data to MDC. Consider the trajectory of MDC in a non-clustered section of WSN in Figure 54 where the MDC enters the communication zone of node 1 and for distance d_1 it remains only in the range of node 1. Similarly, for distance $d_{1,2}$ it is simultaneously in the communication range of node 1 and 2 and for distance $d_{1,2,3}$ it is simultaneously in the communication range of the first three nodes and so on. In this situation, MAC level fairness is thus an important issue since at a given point the MDC may be in the communication range of many nodes. It would be interesting to ensure coordination among these nodes by properly advising a sleep/wake-up pattern so that the available sojourn time is better utilized and MAC level fairness is obtained.

Intelligent routing techniques in WSN with MDC is also planned to be worked on. Let's consider for instance the same scenario where an MDC is collecting data while traversing on a specific path in the network, as shown in Figure 55. It may sometimes be hard or inefficient to adjust the path of the MDC in order to accommodate all sensors in the network to be covered. In this case, sensor nodes lying beyond the communication range of the MDC (such as outside the dotted line in Figure 55) may not get an opportunity to transfer their data to the MDC. However, specific routing techniques can be adopted if the path and speed of the MDC is known. In which cases the out-of-range sensor nodes find a suitable route through some intermediate nodes, which can forward the required data to the MDC.

Last but not least, the evaluation of the proposed schemes in a real testbed is intended to be completed. Therefore, the UAVs under construction at DAS/UFSC will be used.

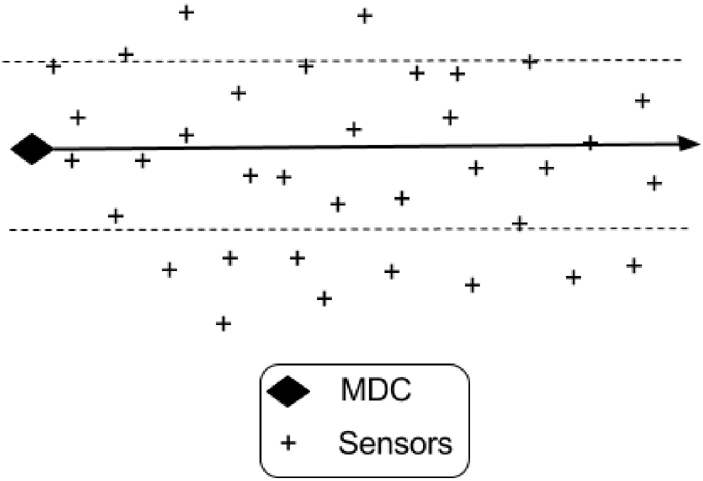


Figure 55 – MDC path in WSN

REFERENCES

ADEWUMI, O. G.; DJOUANI, K.; KURIEN, A. M. Rssi based indoor and outdoor distance estimation for localization in wsn. In: IEEE. **Industrial Technology (ICIT), 2013 IEEE International Conference on**. [S.l.], 2013. p. 1534–1539.

AKYILDIZ, I. F. et al. Wireless sensor networks: A survey. **Comput. Netw.**, Elsevier North-Holland, Inc., New York, NY, USA, v. 38, n. 4, p. 393–422, mar. 2002. ISSN 1389-1286. Disponível em: <http://dx.doi.org/10.1016/S1389-1286(01)00302-4>.

ALI, M.; SULEMAN, T.; UZMI, Z. A. Mmac: A mobility-adaptive, collision-free mac protocol for wireless sensor networks. In: IEEE. **PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.** [S.l.], 2005. p. 401–407.

ALIPPI, C.; VANINI, G. A rssi-based and calibrated centralized localization technique for wireless sensor networks. In: **Proc. IEEE Int. Conference on Pervasive Computing and Communications Workshops (PERCOMW)**. [S.l.: s.n.], 2006. p. 301–306.

AMUNDSON, I.; KOUTSOUKOS, X. D. A survey on localization for mobile wireless sensor networks. In: **Proceedings of the 2Nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments**. Berlin, Heidelberg: Springer-Verlag, 2009. (MELT'09), p. 235–254. ISBN 3-642-04378-X, 978-3-642-04378-9. Disponível em: <http://dl.acm.org/citation.cfm?id=1813141.1813162>.

ANASTASI, G.; CONTI, M.; FRANCESCO, M. D. Extending the lifetime of wireless sensor networks through adaptive sleep. **Industrial Informatics, IEEE Transactions on**, IEEE, v. 5, n. 3, p. 351–365, 2009.

ANASTASI, G.; CONTI, M.; FRANCESCO, M. D. Reliable and energy-efficient data collection in sparse sensor networks with mobile elements. **Perform. Eval.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 66, n. 12, p.

791–810, dez. 2009. ISSN 0166-5316. Disponível em: <http://dx.doi.org/10.1016/j.peva.2009.08.005>.

ANASTASI, G. et al. Energy conservation in wireless sensor networks: A survey. **Ad hoc networks**, Elsevier, v. 7, n. 3, p. 537–568, 2009.

ANEMAET, P. A flexible mac protocol for servicing gossip algorithms. master's thesis. **Delft University of Technology, The Netherlands**, 2008.

ARIAS, J. et al. Malguki: an {RSSI} based ad hoc location algorithm. **Microprocessors and Microsystems**, v. 28, n. 8, p. 403 – 409, 2004. ISSN 0141-9331. Resource Management in Wireless and Adhoc mobile networks. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0141933104000122>.

BACCOUR, N. et al. **Radio link quality estimation in low-power wireless networks**. [S.l.]: Springer, 2013.

BACHIR, A. et al. Mac essentials for wireless sensor networks. **Communications Surveys & Tutorials, IEEE**, IEEE, v. 12, n. 2, p. 222–248, 2010.

BACHRACH, J.; TAYLOR, C. Localization in sensor networks. In: _____. **Handbook of Sensor Networks:Algorithms and Architectures**. John Wiley and Sons, Inc., 2005. p. 277–310. ISBN 9780471744146. Disponível em: <http://dx.doi.org/10.1002/047174414X.ch9>.

BATALIN, M. A. et al. Call and response: experiments in sampling the environment. In: ACM. **Proceedings of the 2nd international conference on Embedded networked sensor systems**. [S.l.], 2004. p. 25–38.

BEEZLEY, B. **K6STI**. FM Radio Stuff, 2008. Disponível em: <http://users.tns.net/ bb/>.

BODANESE, J. P. et al. Infraestrutura de comunicação sem fio para um veículo aéreo não tripulado de curto alcance. 2014.

BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. Gps-less low-cost outdoor localization for very small devices. **Personal Communications, IEEE**, IEEE, v. 7, n. 5, p. 28–34, 2000.

BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. Adaptive beacon placement. In: IEEE. **Distributed Computing Systems, 2001. 21st International Conference on.** [S.l.], 2001. p. 489–498.

CEBIK, L. **W4RNL**. Antennas: Service and Education, 2008. Disponível em: <http://www.cebik.com/>.

CHAKRABARTI, A.; SABHARWAL, A.; AAZHANG, B. Using predictable observer mobility for power efficient design of sensor networks. In: **Proceedings of the 2Nd International Conference on Information Processing in Sensor Networks**. Berlin, Heidelberg: Springer-Verlag, 2003. (IPSN'03), p. 129–145. ISBN 3-540-02111-6. Disponível em: <http://dl.acm.org/citation.cfm?id=1765991.1766001>.

CHAKRABARTI, A.; SABHARWAL, A.; AAZHANG, B. Using predictable observer mobility for power efficient design of sensor networks. In: **Proceedings of the 2Nd International Conference on Information Processing in Sensor Networks**. Berlin, Heidelberg: Springer-Verlag, 2003. (IPSN'03), p. 129–145. ISBN 3-540-02111-6.

CHAKRABARTI, A.; SABHARWAL, A.; AAZHANG, B. Using predictable observer mobility for power efficient design of sensor networks. In: SPRINGER. **Information Processing in Sensor Networks**. [S.l.], 2003. p. 129–145.

CHATZIGIANNAKIS, I.; KINALIS, A.; NIKOLETSEAS, S. Sink mobility protocols for data collection in wireless sensor networks. In: **Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access**. New York, NY, USA: ACM, 2006. (MobiWac '06), p. 52–59. ISBN 1-59593-488-X. Disponível em: <http://doi.acm.org/10.1145/1164783.1164793>.

CHEN, H. et al. Mobile element assisted cooperative localization for wireless sensor networks with obstacles. **Wireless Communications, IEEE Transactions on**, v. 9, n. 3, p. 956–963, March 2010. ISSN 1536-1276.

CHOI, L. et al. M-geocast: Robust and energy-efficient geometric routing for mobile sensor networks. In: **Proceedings of the 6th IFIP WG 10.2 International Workshop on Software Technologies for Embedded and Ubiquitous Systems**. Berlin, Heidelberg: Springer-Verlag, 2008. (SEUS ’08), p. 304–316. ISBN 978-3-540-87784-4.

CHU, M.; HAUSSECKER, H.; ZHAO, F. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. **International Journal of High Performance Computing Applications**, Sage Publications, v. 16, n. 3, p. 293–313, 2002.

CORKE, P.; PETERSON, R.; RUS, D. Coordinating aerial robots and sensor networks for localization and navigation. In: ALAMI, R.; CHATILA, R.; ASAMA, H. (Ed.). **Distributed Autonomous Robotic Systems 6**. [S.l.]: Springer Japan, 2007. p. 295–304. ISBN 978-4-431-35869-5.

DEMIRBAS, M. et al. A fault-local self-stabilizing clustering service for wireless ad hoc networks. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 17, n. 9, p. 912–922, 2006.

DONG, Q.; DARGIE, W. A survey on mobility and mobility-aware mac protocols in wireless sensor networks. **Communications surveys & tutorials, IEEE**, IEEE, v. 15, n. 1, p. 88–100, 2013.

DYO, V. et al. Wildsensing: Design and deployment of a sustainable sensor network for wildlife monitoring. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 8, n. 4, p. 29:1–29:33, set. 2012. ISSN 1550-4859. Disponível em: <http://doi.acm.org/10.1145/2240116.2240118>.

EHSAN, S. et al. Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals. **Wireless Communications, IEEE Transactions on**, v. 11, n. 3, p. 1220–1227, March 2012. ISSN 1536-1276.

FRANCESCO, M. D.; DAS, S. K.; ANASTASI, G. Data collection in wireless sensor networks with mobile elements: A survey. **ACM Trans. Sen. Netw.**, ACM, New York, NY, USA, v. 8, n. 1, p. 7:1–7:31, ago. 2011. ISSN 1550-4859. Disponível em: <http://doi.acm.org/10.1145/1993042.1993049>.

FRANCESCO, M. D. et al. An adaptive strategy for energy-efficient data collection in sparse wireless sensor networks. In: SPRINGER. **European Conference on Wireless Sensor Networks**. [S.l.], 2010. p. 322–337.

GALSTYAN, A. et al. Distributed online localization in sensor networks using a moving target. In: IEEE. **Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on**. [S.l.], 2004. p. 61–70.

GANDHAM, S. et al. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In: **Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE**. [S.l.: s.n.], 2003. v. 1, p. 377–381 Vol.1.

GANESAN, D. et al. Networking issues in wireless sensor networks. **Journal of parallel and Distributed computing**, Elsevier, v. 64, n. 7, p. 799–814, 2004.

GROSSGLAUSER, M.; TSE, D. Mobility increases the capacity of ad hoc wireless networks. **Networking, IEEE/ACM Transactions on**, v. 10, n. 4, p. 477–486, Aug 2002. ISSN 1063-6692.

GU, J.; CHEN, S.; SUN, T. Localization with incompletely paired data in complex wireless sensor network. **Wireless Communications, IEEE Transactions on**, IEEE, v. 10, n. 9, p. 2841–2849, 2011.

GUERRERO, E.; ALVAREZ, J.; RIVERO, L. 3d-adal: A three-dimensional distributed range-free localization algorithm for wireless sensor networks based on unmanned aerial vehicles. In: **Digital Information Management (ICDIM), 2010 Fifth International Conference on**. [S.l.: s.n.], 2010. p. 332–338.

HADIM, S.; MOHAMED, N. Middleware: Middleware challenges and approaches for wireless sensor networks. **IEEE distributed systems online**, v. 7, n. 3, p. 1, 2006.

HAN, G. et al. A mobile anchor assisted localization algorithm based on regular hexagon in wireless sensor networks. **The Scientific World Journal**, Hindawi Publishing Corporation, v. 2014, 2014.

HE, T. et al. Range-free localization schemes for large scale sensor networks. In: **Proceedings of the 9th Annual International Conference on Mobile Computing and Networking**. New York, NY, USA: ACM, 2003. (MobiCom '03), p. 81–95. ISBN 1-58113-753-2. Disponível em: <http://doi.acm.org/10.1145/938985.938995>.

HE, T. et al. Range-free localization schemes for large scale sensor networks. In: ACM. **Proceedings of the 9th annual**

international conference on Mobile computing and networking. [S.l.], 2003. p. 81–95.

HEIDEMANN, J.; YE, W. Energy conservation in sensor networks at the link and network layers. **Wireless Sensor Networks: A Systems Perspective. Artech House Inc**, p. 75–86, 2005.

HEURTEFEUX, K.; VALOIS, F. Is rssi a good choice for localization in wireless sensor network? In: IEEE. **2012 IEEE 26th International Conference on Advanced Information Networking and Applications**. [S.l.], 2012. p. 732–739.

HIGHTOWER, J.; BORRIELLO, G. Location systems for ubiquitous computing. **Computer**, IEEE, n. 8, p. 57–66, 2001.

HILL, J. et al. System architecture directions for networked sensors. **ACM SIGOPS operating systems review**, ACM, v. 34, n. 5, p. 93–104, 2000.

HOESEL, L. van; HAVINGA, P. Collision-free time slot reuse in multi-hop wireless sensor networks. In: IEEE. **2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing**. [S.l.], 2005. p. 101–107.

JARDOSH, A. et al. Towards realistic mobility models for mobile ad hoc networks. In: ACM. **Proceedings of the 9th annual international conference on Mobile computing and networking**. [S.l.], 2003. p. 217–229.

JHUMKA, A.; KULKARNI, S. On the design of mobility-tolerant tdma-based media access control (mac) protocol for mobile sensor networks. In: SPRINGER. **International Conference on Distributed Computing and Internet Technology**. [S.l.], 2007. p. 42–53.

JUANG, P. et al. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In: ACM. **ACM Sigplan Notices**. [S.l.], 2002. v. 37, n. 10, p. 96–107.

JUN, H.; AMMAR, M. H.; ZEGURA, E. W. Power management in delay tolerant networks: a framework and knowledge-based mechanisms. In: **SECON**. [S.l.: s.n.], 2005. v. 5, p. 418–429.

KANNAN, A. A.; MAO, G.; VUCETIC, B. Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In: IEEE. **2006 IEEE 63rd Vehicular Technology Conference**. [S.l.], 2006. v. 2, p. 1022–1026.

KANSAL, A. et al. Power management in energy harvesting sensor networks. **ACM Transactions on Embedded Computing Systems (TECS)**, ACM, v. 6, n. 4, p. 32, 2007.

KANSAL, A. et al. Intelligent fluid infrastructure for embedded networks. In: **Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services**. New York, NY, USA: ACM, 2004. (MobiSys '04), p. 111–124. ISBN 1-58113-793-1. Disponível em: <http://doi.acm.org/10.1145/990064.990080>.

KANSAL, A. et al. Intelligent fluid infrastructure for embedded networks. In: **Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services**. New York, NY, USA: ACM, 2004. (MobiSys '04), p. 111–124. ISBN 1-58113-793-1. Disponível em: <http://doi.acm.org/10.1145/990064.990080>.

KONSTANTOPOULOS, C. et al. Efficient delay-constrained data collection in wireless sensor networks using mobile sinks. In: IEEE. **2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)**. [S.l.], 2015. p. 1–8.

KUROSE, J. F. **Computer Networking: A Top-Down Approach Featuring the Internet, 3/E**. [S.l.]: Pearson Education India, 2005.

LAMBROU, T.; PANAYIOTOU, C. Collaborative event detection using mobile and stationary nodes in sensor networks. In: **Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on**. [S.l.: s.n.], 2007. p. 106–115.

LAMBROU, T.; PANAYIOTOU, C. A survey on routing techniques supporting mobility in sensor networks. In: **Mobile Ad-hoc and Sensor Networks, 2009. MSN '09. 5th International Conference on**. [S.l.: s.n.], 2009. p. 78–85.

LAMBROU, T. P.; PANAYIOTOU, C. G. Collaborative event detection using mobile and stationary nodes in sensor networks. In: IEEE. **Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on**. [S.l.], 2007. p. 106–115.

LEE, J.; CHOI, B.; KIM, E. Novel range-free localization based on multidimensional support vector regression trained in the primal space. **Neural Networks and Learning Systems, IEEE Transactions on**, IEEE, v. 24, n. 7, p. 1099–1113, 2013.

LIU, B. et al. Mobility improves coverage of sensor networks. In: **Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing**. New York, NY, USA: ACM, 2005. (MobiHoc '05), p. 300–308. ISBN 1-59593-004-3. Disponível em: <http://doi.acm.org/10.1145/1062689.1062728>.

LIU, B. et al. Mobility improves coverage of sensor networks. In: ACM. **Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing**. [S.l.], 2005. p. 300–308.

LUNDQUIST, J. D.; CAYAN, D. R.; DETTINGER, M. D. Meteorology and hydrology in yosemite national park: A sensor network application. In: SPRINGER. **Information Processing in Sensor Networks**. [S.l.], 2003. p. 518–528.

LUO, J.; HUBAUX, J.-P. Joint mobility and routing for lifetime elongation in wireless sensor networks. In: IEEE. **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.** [S.l.], 2005. v. 3, p. 1735–1746.

MA, M.; YANG, Y. Sencar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 18, n. 10, p. 1476–1488, 2007.

MARTINEZ, K.; HART, J. K.; ONG, R. Environmental sensor networks. **Computer**, IEEE, v. 37, n. 8, p. 50–56, 2004.

MUNIR, S. A.; BIN, Y. W.; JIAN, M. Efficient minimum cost area localization for wireless sensor network with a mobile sink. In: IEEE. **Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on**. [S.l.], 2007. p. 533–538.

NABI, M. et al. Mcmac: An optimized medium access control protocol for mobile clusters in wireless sensor networks. In: IEEE. **2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)**. [S.l.], 2010. p. 1–9.

NICULESCU, D.; NATH, B. Ad hoc positioning system (aps). In: **Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE**. [S.l.: s.n.], 2001. v. 5, p. 2926–2931 vol.5.

NICULESCU, D.; NATH, B. DV based positioning in ad hoc networks. **Telecommunication Systems**, Springer, v. 22, n. 1-4, p. 267–280, 2003.

OU, C.-H. A localization scheme for wireless sensor networks using mobile anchors with directional antennas. **Sensors Journal, IEEE**, IEEE, v. 11, n. 7, p. 1607–1616, 2011.

OU, C.-H.; HE, W.-L. Path planning algorithm for mobile anchor-based localization in wireless sensor networks. **Sensors Journal, IEEE**, IEEE, v. 13, n. 2, p. 466–475, 2013.

OU, C.-H.; SSU, K.-F. Sensor position determination with flying anchors in three-dimensional wireless sensor networks. **IEEE Transactions on Mobile Computing**, IEEE, v. 7, n. 9, p. 1084–1097, 2008.

PAL, A. Localization algorithms in wireless sensor networks: Current approaches and future challenges. **Network protocols and algorithms**, v. 2, n. 1, p. 45–73, 2010.

PATWARI, N. et al. Locating the nodes: cooperative localization in wireless sensor networks. **Signal Processing Magazine, IEEE**, v. 22, n. 4, p. 54–69, July 2005. ISSN 1053-5888.

PHAM, H.; JHA, S. An adaptive mobility-aware mac protocol for sensor networks (ms-mac). In: IEEE. **Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on**. [S.l.], 2004. p. 558–560.

RABAEY, C. S. J.; LANGENDOEN, K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In: **USENIX technical annual conference**. [S.l.: s.n.], 2002. p. 317–327.

RAGHUNATHAN, V. et al. Energy-aware wireless microsensor networks. **Signal Processing Magazine, IEEE**, IEEE, v. 19, n. 2, p. 40–50, 2002.

SAYYED, A.; ARAÚJO, G. M. de; BECKER, L. B. Smart data collection in large scale sparse wsns. In: IEEE. **2016 9th IFIP**

Wireless and Mobile Networking Conference (WMNC). [S.l.], 2016. p. 1–8.

SAYYED, A. et al. Dual-stack single-radio communication architecture for uav acting as a mobile node to collect data in wsns. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 9, p. 23376–23401, 2015.

SAYYED, A.; BECKER, L. B. Optimizing speed of mobile data collector in wireless sensor network. In: IEEE. **Emerging Technologies (ICET), 2015 International Conference on**. [S.l.], 2015. p. 1–6.

SAYYED, A.; BECKER, L. B. A survey on data collection in mobile wireless sensor networks (mwsns). In: **Cooperative Robots and Sensor Networks 2015**. [S.l.]: Springer, 2015. p. 257–278.

SHAH, R. et al. Data mules: modeling a three-tier architecture for sparse sensor networks. In: **Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on**. [S.l.: s.n.], 2003. p. 30–41.

SHAH, R. C. et al. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. **Ad Hoc Networks**, Elsevier, v. 1, n. 2, p. 215–233, 2003.

SHANG, Y. et al. Localization from mere connectivity. In: ACM. **Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing**. [S.l.], 2003. p. 201–212.

SHRESTHA, N.; YOUN, J.-H.; SHARMA, N. A code-based sleep and wakeup scheduling protocol for low duty cycle sensor networks. In: IEEE. **Networking and Information Technology (ICNIT), 2010 International Conference on**. [S.l.], 2010. p. 80–85.

SICHITIU, M. L.; RAMADURAI, V. Localization of wireless sensor networks with a mobile beacon. In: IEEE. **Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on**. [S.l.], 2004. p. 174–183.

SIMON, G. et al. Sensor network-based countersniper system. In: ACM. **Proceedings of the 2nd international conference on Embedded networked sensor systems**. [S.l.], 2004. p. 1–12.

SINGH, M.; KHILAR, P. M. Mobile beacon based range free localization method for wireless sensor networks. **Wireless Networks**, p. 1–16, 2016. ISSN 1572-8196. Disponível em: <http://dx.doi.org/10.1007/s11276-016-1227-x>.

SMALL, T.; HAAS, Z. J. The shared wireless infostation model: A new ad hoc networking paradigm (or where there is a whale, there is a way). In: **Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing**. New York, NY, USA: ACM, 2003. (MobiHoc '03), p. 233–244. ISBN 1-58113-684-6. Disponível em: <http://doi.acm.org/10.1145/778415.778443>.

SOMASUNDARA, A. et al. Controllably mobile infrastructure for low energy embedded networks. **Mobile Computing, IEEE Transactions on**, v. 5, n. 8, p. 958–973, Aug 2006. ISSN 1536-1233.

SOMASUNDARA, A. A.; RAMAMOORTHY, A.; SRIVASTAVA, M. B. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In: **Proceedings of the 25th IEEE International Real-Time Systems Symposium**. Washington, DC, USA: IEEE Computer Society, 2004. (RTSS '04), p. 296–305. ISBN 0-7695-2247-5. Disponível em: <http://dx.doi.org/10.1109/REAL.2004.31>.

SOMASUNDARA, A. A.; RAMAMOORTHY, A.; SRIVASTAVA, M. B. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In: IEEE. **Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International**. [S.l.], 2004. p. 296–305.

SOMASUNDARA, A. A.; RAMAMOORTHY, A.; SRIVASTAVA, M. B. Mobile element scheduling with dynamic deadlines. **IEEE transactions on Mobile Computing**, IEEE, v. 6, n. 4, p. 395–410, 2007.

SSU, K.-F.; OU, C.-H.; JIAU, H. C. Localization with mobile anchor points in wireless sensor networks. **Vehicular Technology, IEEE Transactions on**, IEEE, v. 54, n. 3, p. 1187–1197, 2005.

STEMM, M. et al. Measuring and reducing energy consumption of network interfaces in hand-held devices. **IEICE transactions on Communications**, The Institute of Electronics, Information and Communication Engineers, v. 80, n. 8, p. 1125–1131, 1997.

SUGIHARA, R.; GUPTA, R. K. Programming models for sensor networks: A survey. **ACM Transactions on Sensor Networks (TOSN)**, ACM, v. 4, n. 2, p. 8, 2008.

SUN, G.-L.; GUO, W. Comparison of distributed localization algorithms for sensor network with a mobile beacon. In: IEEE. **Networking, Sensing and Control, 2004 IEEE International Conference on**. [S.l.], 2004. v. 1, p. 536–540.

SZEWCZYK, R. et al. Habitat monitoring with sensor networks. **Communications of the ACM**, ACM, v. 47, n. 6, p. 34–40, 2004.

TODD, M. et al. A different approach to sensor networking for shm: Remote powering and interrogation with unmanned aerial vehicles. In: **Proceedings of the 6th International workshop on Structural Health Monitoring**. [S.l.: s.n.], 2007.

TONG, H.; HUNG, W.; CHEUNG, C. On-road motor vehicle emissions and fuel consumption in urban driving conditions. **Journal of the Air & Waste Management Association**, Taylor & Francis, v. 50, n. 4, p. 543–554, 2000.

VENKATRAMAN, S.; CAFFERY, J.; YOU, H.-R. A novel toa location algorithm using los range estimation for nlos environments. **Vehicular Technology, IEEE Transactions on**, v. 53, n. 5, p. 1515–1524, Sept 2004. ISSN 0018-9545.

VERMA, A.; SAWANT, H.; TAN, J. Selection and navigation of mobile sensor nodes using a sensor network. **Pervasive and Mobile Computing**, v. 2, n. 1, p. 65 – 84, 2006. ISSN 1574-1192. Disponível em: <http://www.sciencedirect.com/science/article/pii/S157411920500057X>.

VERMA, A.; SAWANT, H.; TAN, J. Selection and navigation of mobile sensor nodes using a sensor network. **Pervasive and Mobile Computing**, Elsevier, v. 2, n. 1, p. 65–84, 2006.

WANG, C. et al. Efficient aerial data collection with uav in large-scale wireless sensor networks. **International Journal of Distributed Sensor Networks**, Taylor & Francis, Inc., v. 2015, p. 2, 2015.

WANG, G. et al. Sensor relocation in mobile sensor networks. In: **INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE**. [S.l.: s.n.], 2005. v. 4, p. 2302–2312 vol. 4. ISSN 0743-166X.

WERNER-ALLEN, G. et al. Deploying a wireless sensor network on an active volcano. **IEEE internet computing**, IEEE, v. 10, n. 2, p. 18–25, 2006.

WHITE, B. et al. Contaminant cloud boundary monitoring using network of uav sensors. **Sensors Journal, IEEE**, v. 8, n. 10, p. 1681–1692, Oct 2008. ISSN 1530-437X.

WILLIG, A. Recent and emerging topics in wireless industrial communications: A selection. **Industrial Informatics, IEEE Transactions on**, IEEE, v. 4, n. 2, p. 102–124, 2008.

XING, G. et al. Rendezvous planning in mobility-assisted wireless sensor networks. In: IEEE. **Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International**. [S.l.], 2007. p. 311–320.

YAN, H. et al. Wireless sensor network based e-health system - implementation and experimental results. **Consumer Electronics, IEEE Transactions on**, v. 56, n. 4, p. 2288–2295, November 2010. ISSN 0098-3063.

ZAIDI, Z. R.; MARK, B. L. Mobility tracking based on autoregressive models. **IEEE Transactions on mobile Computing**, IEEE, v. 10, n. 1, p. 32–43, 2011.

ZHIYONG, T.; DARGIE, W. A mobility-aware medium access control protocol for wireless sensor networks. In: IEEE. **2010 IEEE Globecom Workshops**. [S.l.], 2010. p. 109–114.