Peripheral Routing Protocol – a new routing protocol proposal for a realistic WSN mobility model

by

Ion Gabriel Tudorache

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

University of Leeds

School of Electronic and Electrical Engineering

August 2014

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgment.

a. Chapter 2, chapter 3 (section 3.1) and chapter 4 are based on work from jointly authored publications, whose main author is I.G. Tudorache.

b. Details of the publications which have been used (e.g. titles, journals, dates, names of authors):

In Chpater 2: “Improved mesh wsn support for a realistic mobility model.” In Wireless Communication Systems (ISWCS), 2010 7th International Symposium on, pages 340–344, 2010, I.G. Tudorache, A.M. Popescu, and A.H. Kemp.

In Chapter 3, section 3.1: “Surveying position based routing protocols for wireless sensor and ad-hoc networks”. International Journal of Communication Networks and Information Security (IJCNIS), Vol. 4, No. 1:41–67, April 2012, Bo Peng A.H. Kemp Ana Maria Popescu, Ion Gabriel Tudorache.

In Chapter 4: "Manet routing protocols problem for the marginal mobility model." In Microwave Conference (EuMC), 2011 41st European, pages 139–142, 2011, I.G. Tudorache, A.M. Popescu, and A.H. Kemp.

c. Details of the work contained within these publications which is directly attributable to I. G. Tudorache:

With the exceptions detailed in section d., the published work is entirely attributable to I.G. Tudorache: the literature review necessary to develop the ideas behind the published manuscripts, the novel ideas proposed in the papers, the development of the mathematical model and MATLAB WSN simulator used in the analysis and all the work necessary in the editing of the manuscripts.

d. Details of the contributions of the other authors to the work:

Dr. A. H. Kemp is the co-author for all the publications listed above. They have been written under his supervision, benefiting from excellent technical and editorial advice.

A.M. Popescu has provided information about routing protocols in WSNs and aided with critical thinking about simulation scenario set-up.

To my family:

my parents.... Mr. Dumitru Tudorache and Mrs. Elena Tudorache,

my brother....Mihai Constantin Tudorache,

my loved one....Ioana Loredana Mihai,

Acknowledgements

I would like to express my gratitude towards my supervisor Dr. A. H. Kemp for the useful comments, knowledge, patience, motivation, enthusiasm and immense support during the four years of study at University of Leeds for my PhD study and research. Regular meetings and presentations held with him helped me improve my understating about wireless sensor networks challenges which lead to my thesis subject. His guidance to question everything and always have a reasoned opinion lead to a better and clear understanding of my entire study.

Furthermore I would like to thank all my colleagues from the WSN group for the regular meetings, presentations and support. Their enthusiasm and critical opinions helped me improve my work and enhance my horizon with different perspectives. They were always available for discussions and their advises were welcomed and much appreciated. I am so glad that I could be a part of this fantastic group.

I would also like to thank my family, loved ones and friends who have supported me throughout the entire process, both by keeping me harmonious and helping me putting pieces together. Family and friends support is very important throughout such a difficult and challenging process so I consider myself a very lucky person.

Abstract

Wireless sensor networks (WSNs) are changing our way of life just as the internet has revolutionized the way people communicate with each other. Future wireless networks are envisioned to be robust, have simple and efficient communication between nodes and self-organizing dynamic capabilities. When new nodes join in, a self-configuring network has to have the ability to include these nodes in its structure in real time, without human or machine interference. The need for a destination node (D) which moves at the periphery of wireless sensor networks can be argued from different points of view: the first is that different WSN scenarios require data gathering in such a way; the second point is that this type of node movement maximizes network lifetime because it offers path diversity preventing the case where the same routes are used excessively. However the peripheral movement model of the mobile destination does not resemble any mobility models presented in the WSN literature. In this thesis a new realistic WSN sink mobility model entitled the "Marginal Mobility Model" (MMM) is proposed. This was introduced for the case when the dynamic destination (D), moving at the periphery, frequently exits and enters the WSN coverage area. We proved through Qualnet simulations that current routing protocols recommended for Mobile Ad Hoc Networks (MANETs) do not support this sink mobility model. Because of this, a new routing protocol is proposed to support it called the Peripheral Routing Protocol (PRP). It will be proven through

MATLAB simulations that, for a military application scenario where D's connectivity to the WSN varies between 10%-95%, compared with the 100% case, PRP outperforms routing protocols recommended for MANETs in terms of throughput (T), average end to end delay (AETED) and energy per transmitted packet (E). Also a comparison will be made between PRP and Location-Aided Routing (LAR) performance when D follows the MMM. Analytical models for both PRP and LAR are proposed for T and E. It is proved through MATLAB simulations that, when compared with LAR, PRP obtains better results for the following scenarios: when the WSN size in length and width is increased to 8000 m and one packet is on the fly between sender and sink, PRP sends 103% more data and uses 84% less energy; when more data packets are on the fly between sender and sink, PRP sends with 99.6% more data packets and uses 81% less energy; when the WSN density is increased to 10,000 nodes PRP uses 97.5% less energy; when D's speed in increased to 50 Km/h, PRP sends 74.7% more data packets and uses 88.4% less energy.

Contents

List of Figures

List of Tables

List of Abbreviations

AETED	Average End to End Delay
AODV	Ad Hoc On-Demand Vector
D	Destination
DIN	Delay Intolerant Network
DLP	Destination Location Prediction
DSR	Dynamic Source Routing
DTMSN	Delay Tolerant Mobile Sensor Network
DTN	Delay Tolerant Network
DYMO	Dynamic On-Demand
E	Energy per transmitted packet
EAGPR	Energy Aware Geographic Routing Protocol
GPRS	Greedy Perimeter Stateless Routing
GPS	Global Positioning System
IGF	Implicit Geographic Forwarding

ISM	Industrial Scientific and Medical
LAN	Local Area Networks
LAR	Location Aided Routing
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MMM	Marginal Mobility Model
MRLG	Mobile-sink Routing for Large Grids
NE	North-East
NS	Network Simulator
NW	North-West
P	Peripheral Nodes
Pckts	Packets received
PDR	Packet Delivery Ratio
PHY	Physical
PRA	Peripheral Routing Algorithm
PRP	Peripheral Routing Protocol
R	Range
RERR	Route Error
RREP	Route Reply

RREQ	Route Request
S	Source
SE	South-East
SEAD	Scalable Energy-efficient Asynchronous Dissemination
SW	South-West
T	Throughput
TRAMA	Traffic Adaptive Medium Access Protocol
TTDD	Two-Tier Data Dissemination
VNE	Virtual Network Environment
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Networks
WMN	Wireless Mesh Network
WPAN	Wireless Personal Area Networks
WSAN	Wireless Sensor and Actor Networks
WSN	Wireless Sensor Network

List of Symbols

Chapter 2

S	source node
P_N	north WSN peripheral nodes
P_S	south WSN peripheral nodes
P_E	east WSN peripheral nodes
P_W	west WSN peripheral nodes
N_1	peripheral node
x_a, y_a	x and y coordinates of node a
x_m, y_m	x and y coordinates of the sink
r	communication range
$C_i(x_{ci}, y_{ci})$	check points on the pathway

Chapter 4

D	destination
(X_d, Y_d)	coordinates of D

Chapter 5

N_{WSN}	all the nodes from the WSN area
N_i	node i from the WSN area
N_a	node a from the WSN area
N_b	node b from the WSN area

x_b, y_b	x and y coordinates of node b
N_1, N_2, N_3	node 1, 2 and 3 from the WSN area
ρ	perimeter length for neighbours check
C_F	connectivity factor
T_C	connectivity time
T_D	disconnectivity time
C_T	connectivity threshold factor
P	WSN peripheral nodes
X_i	connectivity indicator between a peripheral node i and D
PCV	peripheral connectivity vector
PCV_N	north peripheral connectivity vector
PCV_S	south peripheral connectivity vector
PCV_E	east peripheral connectivity vector
PCV_W	west peripheral connectivity vector
PCV_N^b	binary value for PCV_N
PCV_S^b	binary value for PCV_S
PCV_E^b	binary value for PCV_E
PCV_W^b	binary value for PCV_W
PCV_N^d	decimal value for PCV_N^b
PCV_S^d	decimal value for PCV_S^b
PCV_E^d	decimal value for PCV_E^b
PCV_W^d	decimal value for PCV_W^b
D_m	direction of movement indicator
$P_{\alpha_{t_\beta}}$	probability of switching direction at t_β
$M_{x_{t_\beta}}$	the sum of the connectivity vectors generated at different periods of time t_i, $i = 1, \beta$

α_{t_β}	$M_{x_{t_\beta}}$ highest value
P_{α_r}	probability of switching direction threshold
d	distance that D travels disconnected between two peripheral nodes
v	D speed
l	D direction of movement
$N_{(X_N,Y_N)}$	current D neighbour
$M_{(X_M,Y_M)}$	D next neighbour
$A_{(X_A,Y_A)}$	D entry point in N communication range
$B_{(X_B,Y_B)}$	D exit point from N communication range
$C_{(X_C,Y_C)}$	D entry point in M communication range
$E_{(X_E,Y_E)}$	D exit point from M communication range
m	l slope
$C(O,R)$	circle C of center O and range R

Chapter 6

T	throughput
E	energy spent per transmitted packet
T_{sim}	simulation time
P_{ckts}	total number of packets received
$Pckts_{LAR}$	total number of packets sent by S to D when LAR is used
n_p	path length between S and D
t_s	time between two data packets sent by S
t_p	time to send a data packets between two nodes which are one hop away
p_r	packet rate
d_r	data rate
p_s	packet size

T_{RC}	connectivity time between D and its neighbour in which D receives packets from S
T_{RS}	time in which D receives packets directly from S after its neighbour buffer gets empty
T_{IC}	ideal (maximum) connectivity time between D and its neighbour
τ_L	time that D travels connected with its neighbour until it receives a RREQ packet from S when LAR is used
τ_P	time that D travels connected with its neighbour until it receives a RREQ packet from S when PRP is used is used
T_{RREP}	time spent by D to send a RREP packet to S
ξ_L	LAR RREQ broadcast time window
ξ_P	PRP RREQ broadcast time window
T_{RREQ}	time to send a RREQ from S to the furthest node from the RREQ zone
T_{wait}	time between two RREQ for LAR and time between two hello packets for PRP
R_i	positions where D is when a RREQ is sent at each ξ seconds
R_5	position where D is found in M communication range
R_6 (for LAR)	position where D starts to receive data packets from S
T_{hello}	time spent by D between two hello packets
T_{IS}	time spent by M to inform S to stop and resend data packets
R_6 (for PRP)	position where D starts to receive data packets from M buffer
R_8 (for PRP)	position where D starts to receive data packets from S
T_{RB}	time spent by M to send all the packets from its buffer to D
$Pckts_b$	packets sent from M's buffer
$Pckts_c$	packets sent durring T_{RS}

T_{RS}	is the time at each D receives packets from S each t_s seconds
$Pckts_{PRP}$	total number of packets sent by S to D when PRP is used
E_{LAR}	total energy spent by LAR
$E_{LAR_{Routing}}$	energy spent by LAR on routing
$E_{LAR_{RREQ}}$	energy spent by LAR on RREQ packets
e_{rreq}	energy spent on one RREQ between two neighbour nodes
n_p	path length between S and D
n_{rreq}	number of route requests sent in the network
$e_{routing}$	energy spent to send a data packet between two nodes which are one hop away
E_{PRP}	total energy spent by PRP
$E_{PRP_{Routing}}$	energy spent by PRP on routing
$E_{PRP_{hello}}$	energy spent by PRP on hello packets
$E_{S_{RREQ}}$	energy spent by S to find a path to D next neighbour
n_{hello}	numbe of hello packets sent in the network
n_n	total number of nodes from the RREQ area
n_m	total number of nodes which are in an area defined by D and S coordinates
$\hat{d}$	distance approximation with location error
ϵ	location error
$\hat{T}_D$	T_D affected by the location errors
θ	confidence time interval
T_B	time after which D starts the broadcast of hello packets when location error is considered
$\triangle$	distance associated to θ
C_1	C with location errors

C_2	the point where D starts to send hello packets with θ seconds sooner
$d(\hat{N, B})$	estimated distance between N and B
$d(\hat{N, C_2})$	estimated distance between N and C_2
$d\,(C, E)$	distance between C and E
T_{error}	time lost from the connectivity due to location errors
$Pckts_{PRP_E}$	total packets sent by PRP in presence of location errors

1 Wireless sensor networks

1.1 Introduction

Wireless Sensor Networks (WSN) are changing our way of life just as the internet has revolutionized the way people communicate with each other [1]. A wireless sensor network is a network made primarily of wireless sensor nodes that communicate to meet the objective for which the network was designed. Each wireless sensor node has the ability to perform different tasks such as: gathering sensory information from the medium (detect movement, temperature, humidity, radiation, etc.), process this information and communicate with the other sensor nodes from the network (sending or forwarding this gathered information). A wireless sensor network is a network in which each sensor node supports a multi-hop routing algorithm, meaning that several nodes can collaborate to forward data to the base station. An example of a WSN structure is presented in Figure 1.1 . Each wireless sensor node has the following components: one or more sensors; radio transceiver or other wireless communication device (laser communication for example); a small micro controller and a energy source (usually a battery).

The dimension of a sensor node can vary from a box of few centimeters to the size of a grain of dust, although currently functioning nodes of such small size are only a concept. The cost of a sensor node depends on the size and complexity and varies

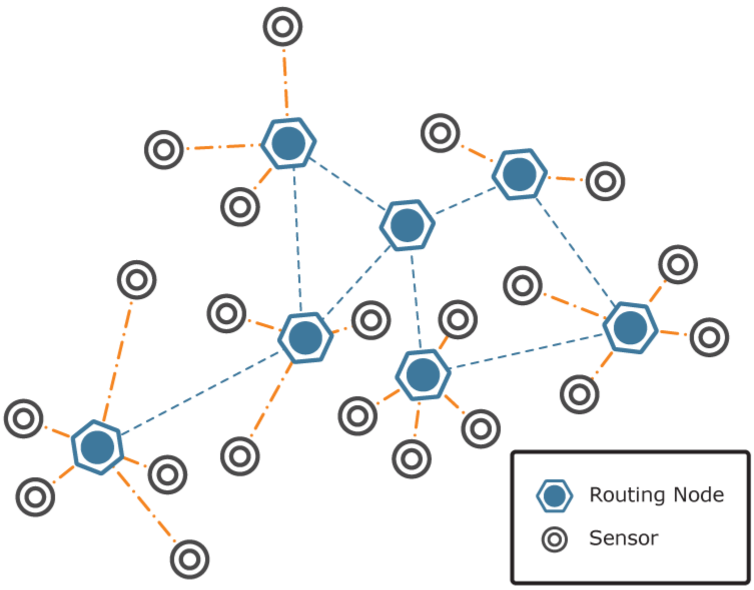

Figure 1.1: WSN structure
[2]

from a few pennies to hundreds of dollars. These two main characteristics, size and cost, have a crucial effect on resource constrains that a sensor node has, i.e. on energy, memory, computational speed and bandwidth [3]. An example of a wireless sensor node is presented in Figure 1.2 .

Future wireless networks are envisioned to be robust, have simple and efficient communication between nodes and self organizing dynamic capabilities. When new nodes join in, a self configuring network needs to have the ability to include these nodes in its structure in real time, without human or machine interference. In this way the complexity of the network deployment process and the network maintenance are reduced leading to minimal upfront investment [5]. Wireless Mesh Networks (WMNs), due to their design, promise to satisfy the demands of an optimal sensor network, require minimal investment and make possible the communication anywhere, anytime and anyhow. So, these are the types of networks which will dominate the future in wireless local area networks (WLANs), wireless personal

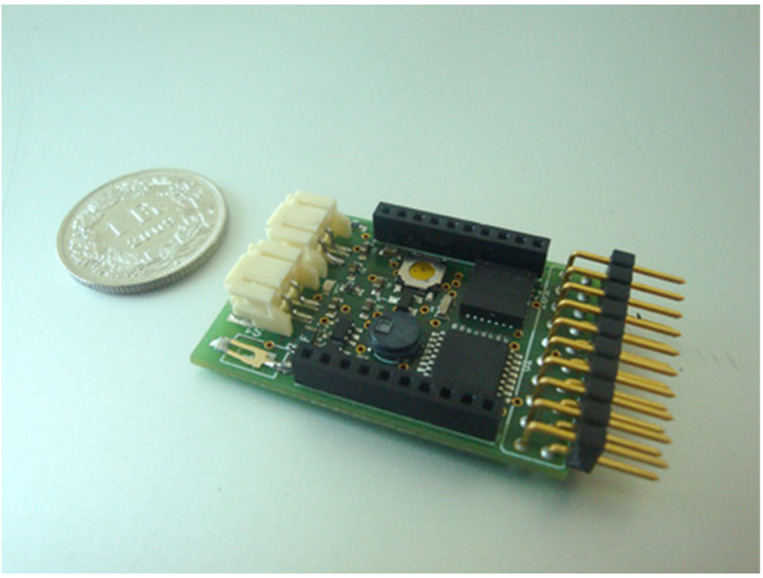

Figure 1.2: WSN hardware sample
[4]

area networks (WPANs), wireless sensor networks (WSNs) or wireless metropolitan area networks (WMANs) [6].

In the last few years, WMNs have evolved rapidly. Research in this area has resulted in several companies currently offering different products based on WMNs. Despite all the efforts made to develop this new promising technology there are still many issues to consider: the available MAC and routing protocols recommended for WMNs are not scalable enough; WMNs are supposed to grow from few hundreds of nodes to millions or even more but with the current recommended technology the throughput drops significantly as the number of nodes or hops increases. The open challenge is to enhance or come up with new solutions from the application layer, to transport, network, Medium Access Control (MAC) and physical layers [5].

WSNs applications are presented in the next section. The role of presenting these applications is to:

- Understand all the areas in which the WSNs can be used.

- Get familiar with the many environments in which the wireless sensors are needed.
- Observe that, depending on the application, special constrains can apply to the WSNs.

1.2 WSN applications

WSNs are used in many different areas of applications, e.g. area monitoring, environmental monitoring, industrial monitoring, etc. Different WSNs applications are presented further:

1. **Military applications** – one of the primary uses for WSNs in this area would be remote sensing. A survey of WSN military applications is made in [7]. An example of a battle field can be considered where the enemy intrusion needs to be monitored Figure 1.3 . For this a first step will be to deploy wireless sensor nodes across the monitored area using different techniques (by air using airplanes or by land using specially built robots) [8], [9]**.** These sensors will form a WSN that will cover the entire battlefield and that can monitor different parameters like heat, pressure, sound, light, electromagnetic field, vibrations, etc.. The U.S. Army's Disposable Sensor Program is an initiative by the U.S. Army to deploy nodes with seismic, acoustic, RF, magnetic, chemical, biological, and infrared sensors [10]. Through these sensor nodes, the army will obtain a complete picture of almost any environment. These sensors would have to be disposable and inexpensive. Urban warfare is another application in which WSN can be used. Nodes could be deployed in a urban environment with the purpose to detect enemy intrusion, movement or chemical attacks [11]. The collected information from the WSN area can be sent to a static base or

mobile military unit which moves on the WSN periphery. This information can be vital in saving lives or to reduce material loses so it needs to reach the destination fast.

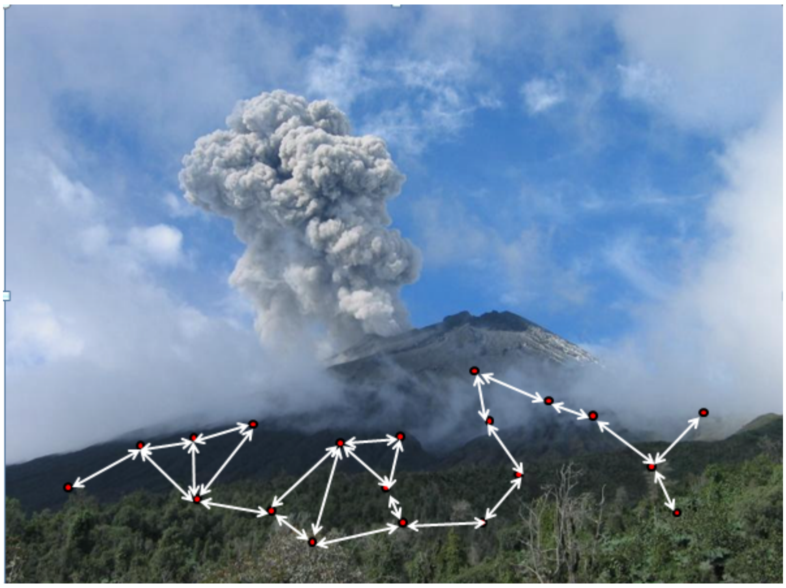

Figure 1.3: WSN military application scenario [12]

2. **Building monitoring** - it is expected that in the near future each building will be monitored through the use of a WSN as a standard functionality. The sensors will monitor heat and smoke which allows detection, localization and tracking of fires. The information gathered from the building can help occupants by guiding them to the nearest exit and also fireman's to decide which will be the best approach to avoid a disaster [13]. Smart Buildings are envisioned to have smart nodes which could allow distributed monitoring and control to improve the living conditions and reduce to minimum the energy consumption [11].

   Examples from literature on how the energy consumption can be reduced in smart buildings through the use of WSN are [14], [15], [16], [17]:

- Lights, chairs and tables automatically adjust as soon as the family gathers in the living room to watch TV.
- Phones only ring in rooms where someone is present.
- Music in a room adapts automatically to people within.
- Support elderly people in their daily life routines.
- Security systems like emergency alarms for burglars, fire with complete awareness of home owners wherever they are.
- Natural interface of devices through speech and gesture.

Through the use of WSN, the headquarters building of the New York Times consumes 30% less energy than traditional office buildings [18]. This building system relies on a large-scale sensor network which gathers information from the offices and uses air conditioning, heating, etc. only when needed. In an event of fire, firefighters outside the building can receive real time information which can help them minimize the loss of lives and material. So, receiving this information fast is crucial in these type of situations.

3. **Habitat monitoring -** the deployment of sensor nodes provides a non-invasive approach for environments which are rather sensitive to human presence [11], [19]. For example, through the use of sensors, environmental data (air temperature, light, wind, relative humidity and rainfall) is gathered in the South-West Rift Zone, in Volcanoes National Park on the Big Island of Hawaii for the Pods project at the University of Hawaii at Manoa [20]. Another example is the PermaSense Project [21] which was designed to gather reliable data of physical parameters in natural and diverse slope areas in Swiss Alps Figure 1.4. With this data they aim to develop theoretical models for temperature simulation and for hazard assessment.

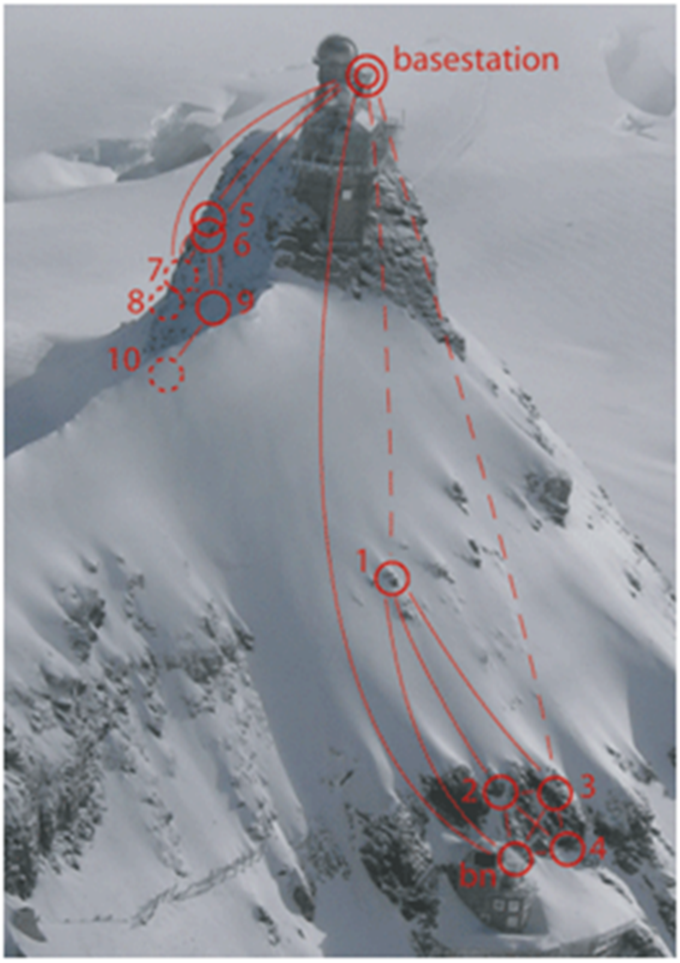

Figure 1.4: WSN PermaSense project
[21]

In Princeton's Zebranet Project [22], sensor nodes are used to monitor zebras movement and behavior. It is crucial not to have human presence for wildlife monitoring in order to obtain accurate data which represents animals behavior in natural conditions. These areas are usually large so data collection can be made with the help of mobile units which move outside the WSN area, on the boundary. In [18] a plane deploys 30 airborne sensors in the arctic for analysis and assessment of global warming. The collected information can be sent to a boat or a plane which surpasses at a low altitude the monitored area.

4. **Industrial applications -** these can require networks to function in both in-door and outdoor environment. Possible applications refer to monitoring and control of industrial equipment, processes and personnel. Few examples of WSN industrial applications are given by Banner Engineering [**23**] as follows:

   - Wireless perimeter guard - by using a solar powered wireless alert system they can alert the security personnel when a security breach was detected Figure 1.5.
   - Monitoring in a bottling facility - a bottling plant monitors the level, pressure and temperature inside each rotary filler to accurately determine when to activate the inflow into the tank. WSNs offer a viable solution to eliminate the wires on moving assemblies (Figure 1.6).
   - Day tank level monitoring - for tank locations not wired for power or data transfer, a wireless sensor network can gather the tank data and transmit it to a central control location for analysis and logging. Day tank fill levels are monitored continuously so that materials are maintained at a specific quantity. Because these tanks are slowly filled or depleted, ultrasonic sensors can be used to measure the tank levels without false readings or tank wall interference (Figure 1.7).

WSN specifications and constrains are presented further in order to understand the challenges that these impose.

1.3 WSN specifications and constraints

WSN are are different from ad-hoc Wireless Networks and have special characteristics like [24]:

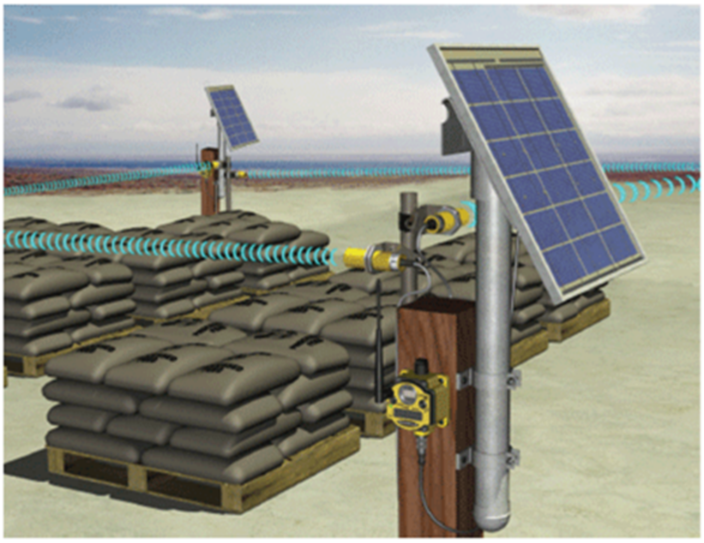

Figure 1.5: Wireless perimeter guard
[**23**]

- The topology of a sensor network changes frequently in comparison with the one of an ad-hoc network.
- The number of sensor nodes can be several orders of magnitude higher in a sensor network than the nodes in an ad-hoc network.
- In a sensor network the nodes are densely deployed.
- In a sensor network the sensor nodes have restrictions of power, memory and computational capacities.
- Sensor nodes are prone to failure.
- Because of the large size of a sensor network, a node may not have a global identification number.

WSNs are used in a variety of applications domains, as it was shown in the previous section, and each scenario may require: collaborative sensing, communication, and computation among multiple sensors that observe moving objects, physical effects and/or environmental events. Also, each scenario is commonly structured in tasks

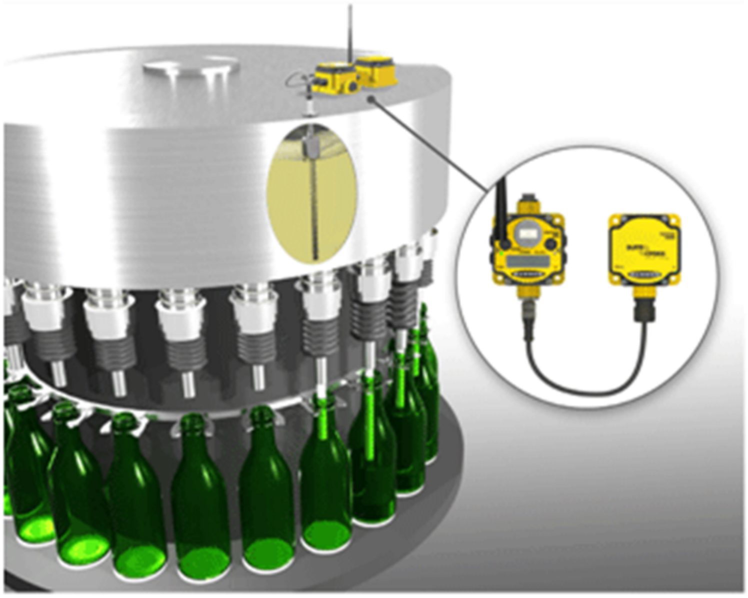

Figure 1.6: WSN monitoring in a bottling facility
[**23**]

such as: deployment, application functionality and information exchange [25]. These tasks can affect the network proprieties. For example, the deployment task affects network proprieties such as node density and topology. It can also predetermine the data collection and routing algorithms by providing connectivity degree and sensing coverage. Coverage means that every location in the sensing field is monitored by at least one sensor. A network has a coverage degree k if every location is within the sensing range of at least k sensors. Connectivity means that any node can communicate with any node from the network using intermediary nodes if the direct communication is not possible. Sensing and communications range help maintaining coverage and connectivity. Choosing a deterministic sensor placement in WSN deployment can help improve the network proprieties but there are applications in which this is not possible, like: military applications, remote areas and hazardous environments. Here a random sensor distribution is used.

Figure 1.7: WSN day tank level monitoring
[**23**]

Besides improving the coverage and connectivity, another deployment objective is the trade-off between the network lifetime and the number of sensor nodes. Different deployment strategies can increase the network efficiency with respect to energy, cost, speed and life time. Many aspects need to be taken into consideration when choosing to design a wireless sensor network application.

Depending on the application, the network design has different constrains and is influenced by factors like: fault tolerance; scalability; production costs; operating environment; sensor network topology; hardware constraints; transmission media; and power consumption [26]. These factors are explained as follows:

1. **Fault tolerance** - is the ability to sustain sensor network functionality without any interruption due to sensor node failures [27]. Depending on the application for which the sensor network is designed, a different level of fault tolerance can be considered. For example, if the environment where the sensors are deployed has great interference then the sensor network needs to be

designed for high fault tolerance. An example is a sensor network deployed in a battlefield for a military application.

2. **Scalability -** depending on the application for which the sensor network is designed, the number of sensor nodes can vary from tens to millions. The communication between sensor nodes does not need to be influenced by the sensor nodes density in different regions of the network. The production cost needs to be maintained as low as possible when designing a sensor network application: The price for each sensor node needs to be as low as possible and because of this each sensor node needs to have a low degree of complexity. Also both upfront investment and maintenance cost need to be as low as possible.

3. **Hardware constrains** – a sensor node contains four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit. Depending of the application for which are designed new units can be added, like a Global Positioning System (GPS). The most important unit is the power unit because the lifetime of a sensor node depends directly on the lifetime of its battery. A sensor node needs to be very small and apart from this there are stringent constrains like: consume extremely low power; have low production cost; to be autonomous and adaptive to the environment.

4. **Power consumption** – in many applications replacing the sensor battery is impossible thus the life time of a sensor node is constrained by the life time of its battery. Depending on the application a different metric can be used to characterize the efficiency like power, delay, throughput, etc. Each sensor nodes consumes power when one of the following actions is made: sensing, data communication or data processing. Energy spent for data processing is much less when compared with data transmission, so data processing is very efficient when it comes to save power. For example, the same amount of energy is used

to send 1 Kb of data over a 100 m range or to execute 3 million instructions by a 100 million instructions per second processor.

5. **Sensor network topology** – In a sensor network application nodes can be deployed in close proximity with each other. The nodes can be deployed by various methods, depending on the application for which they are meant. For a military application, for example, they can be dropped from planes or placed one by one by a robot. After the sensor deployment the network topology may change due different factors like:

    - Energy level – each sensor has a limited power energy which will end. Because of this different sensor nodes at different periods of time will become unavailable which will lead to network topology changes.
    - Communication range – due to the energy level, each sensor's communication range will decrease in time. Also the communication range can be affected by different obstacles or interferences that can appear in time. All of this will lead to network topology changes in the original network structure.
    - Faulty sensors – network topology changes can appear due to faulty sensors.
    - Mobility – for certain applications some sensors are mobile thus the network topology changes.
    - New nodes can be added after the initial deployment which can lead to changes in the network topology.

The communication between sensor nodes can be made through radio, infrared or optical media:

- For radio links is recommended to use the industrial, scientific and medical bands (ISM). The ISM bands are not bound to particular standards so different power saving communication protocols can be used. Interferences from existing applications can cause an important drawback.

- Infrared communication is robust to interferences from existing applications but the main disadvantage is the line of sight requirement for communication.
- Optical communication advantages are [28]: optical devices are smaller and consume less power; reflection, diffraction, scattering from aerosols help distribute signal over large areas and optical wireless provides freedom from interference and eavesdropping within an opaque enclosure. Some important drawbacks of optical communication in WSN are: the optical signal has to be contained within the field-of-view of the transmitter and receiver and due to the random nature of deployment of sensors, this cannot be always fully ensured; bandwidth reduces in non-line-of-sight links, due to multipath propagation; increasing the field-of-view of the transceiver system, on the other hand, implies increase in transmit power to maintain a viable link and also increase in the interference plus noise floor for neighboring communicators.

For a military application the communication can encounter greater interference thus the choice of the transmission medium needs to be supported by robust coding and modulation scheme.

In the next section the motivation, novelty and contribution of this thesis are presented.

1.4 Motivation, novelty and contribution

This section states the scope and motivation behind this research in WSNs.

1. **Mobility Models used in WSN simulations:**

a) Motivation: Chapter two presents the research made in the areas of mobility models used in WSN simulations. A mobility model should represent a realistic behavior of mobile nodes in a mobile ad-hoc network. The authors in [29] sustain that mobility models directly impact the performance of routing protocols so using an unrealistic mobility model will lead to inaccurate performance measurements. In [30] it is stated that past evaluations of multicast routing protocols have utilized a single, simple random way point model e.g. [31], [32], [33] and [34]. So the majority of existing mobility models for ad-hoc networks do not capture the variety of mobility patterns likely to be exhibited by ad-hoc applications.

b) Novelty and Contribution: A military application scenario is considered in which the mobile sink throughout its movement joins and leaves the WSN communication area frequently. This sink mobility model is called the "Marginal Mobility Model" (MMM). The MMM is a realistic mobility model in which a mobile sink moves at the WSN periphery. Because of this type of movement, current routing protocols proposed for MANETs will experience difficulties in sending data packets to a sink which follows the MMM. The MMM is introduced in "Improved Mesh WSN Support For A Realistic Mobility Model" [8] and discussed further in "MANET Routing Protocols Problem For The Marginal Mobility Model" [35].

2. **WSN Routing Protocols:**

a) Motivation: Chapter three presents the research made in the area of WSN routing protocols. Routing in WSN is very challenging due to several characteristics that distinguish them from contemporary communication and wireless ad-hoc networks. Routing protocols need to consider the characteristics of sensor nodes, application and architecture requirements.

The performance of a routing protocol is closely related to the architectural model which is directly influenced by the application requirements. Routing performances are also influenced by important design parameters such as: loop free, distributed operation, path strategy, packet forwarding, path selection metrics, memory (state), guaranteed message delivery, scalability, overhead and adaptive to mobility.

b) Novelty and Contribution: A journal paper entitled “Surveying Position Based Routing Protocols for Wireless Sensor and Ad-ho Networks” [36] was published which is a survey of almost 50 position-based routing protocols and it comes as an aid in the implementation of this type of routing in various applications which may need to consider the advantages and pitfalls of position-based routing. An emphasis is made on geographic routing, whose notion is clarified as a more restrictive and more efficient type of position-based routing. The protocols are therefore divided into geographic and non-geographic routing protocols and each is characterized according to a number of network design issues and presented in a comparative manner from multiple points of view. The main requirements of current general applications are also studied and, depending on these, the survey proposes a number of protocols for use in particular application areas. This aims to help both researchers and potential users assess and choose the protocol best suited to their interest.

3. **MANETs routing protocols support of the MMM:**

a) Motivation: The need for a destination node (D) which moves at the WSN periphery can be argued from different points of view: application requirements - certain WSN scenarios require data gathering in such a way; energy saving - the sink movement maximizes network lifetime offering path diversity and preventing the case where the same routes are used excessively. Numerous routing protocols have been proposed for ad-hoc mobile networks and five of these are presented in chapter five.

b) Novelty and Contribution: The conference paper entitled “MANET Routing Protocols Problem For The Marginal Mobility Model” [35] proves through simulations that five routing protocols recommended for MANETs do not support a sink movement. A WSN scenario is considered using Qualnet simulator in which the mobile node follows the MMM. It is shown that routing protocols like LAR (Location-Aided Routing)[37], DSR (Dynamic Source Routing) [38], AODV (Ad Hoc On-Demand Vector) [39], Bellman Ford and DYMO (Dynamic On-Demand) [40], [41], [42] cannot send data packets to the mobile node. Different simulations are made by varying D speed and distance from the WSN coverage area.

4. **The peripheral routing protocol (PRP):**

a) Motivation: Multiple WSN applications require the presence of a mobile sink at the network periphery to receive/collect data. The area on which the sensors are deployed might be: inaccessible to humans (toxic or radioactive); inaccessible to mobile units (rough terrain); sensitive to any kind of intrusion (natural wildlife reservations). Also, in time the number of void areas (areas which are not covered by any sensor node) in the WSN might increase due to sensors which will remain without power. A mobile sink at the WSN periphery during its movement will join and leave the WSN coverage area frequently. The same thing will happen if the mobile node will move through a WSN in which the number of voids area is high. This type of mobility model was introduced as the MMM. A mobile node can follow the MMM also inside a WSN where voids are common and where throughout its movement, the mobile node will pass frequently through these voids. Also, the sink movement at the WSN periphery leads to path diversity and energy consumption distribution among a high number of nodes from the network. Thus, the life time of a WSN is improved with this type of movement. It is difficult to find a well-connected end-to-end path for all the sensors to transmit data to D when it follows the MMM because D will enter and exit the WSN coverage area frequently. This case is most often met in DTMSN networks. Delay tolerant networks (DTNs) embrace the concept of occasionally-connected networks that may suffer from frequent partitions [43]. Using a DTMSN routing protocol is not a solution even if the WSN has DTMSN characteristics. The reason is that DTMSN routing protocols require from the application itself the DTN characteristics to set up the routing protocol. However, for the proposed military application scenario, the DTN characteristics are not known nor are they triggered by the application but by the WSN scenario. Also, it was shown through Qualnet simulations that MANET routing protocols do not support this mobility model.

b) Novelty and Contribution: A new routing protocol which supports the MMM is described in chapter five. It is shown through MATLAB simulations that, for a military application scenario where D's connectivity to the WSN varies between 10%-95%, compared with the 100% case, PRP outperforms routing protocols recommended for Mobile Ad-ho Networks (MANET) in terms of throughput, average end to end delay and energy per transmitted packet. In chapter five of this thesis the PRP is described. An analytical model for the throughput and energy for both PRP and LAR is presented in chapter six. In chapter seven, through

MATLAB simulations, it was shown that PRP outperforms LAR performances when the sink follows the MMM and when WSN size, density and D's speed are varied.

The thesis outline is presented in the next section.

1.5 Thesis outline

This thesis comprises of eight chapters as follows:

- **Chapter 1:** A general introduction regarding WSN is made in section 1.1. Section 1.2 presents different WSN applications. WSN specifications and constraints are made in section 1.3 and thesis motivation, novelty and contribution are discussed in section 1.4. This chapter ends with the thesis outline in section 1.5.
- **Chapter 2:** The importance of mobile sinks in WSNs is discussed in chapter two. In section 2.1 different mobility models used in WSNs are presented and in section 2.2. the Marginal Mobility Model (MMM) is introduced. This chapter ends with a presentation of the MMM's characteristics.
- **Chapter 3:** The sink's mobility support in WSNs is analysed in this chapter as follows: section 3.1 presents routing protocols classifications and specifications; routing protocols proposed to support mobile sinks are presented and discussed in section 3.2 analysing if the MMM can be supported; section 3.3 presents location prediction schemes used for mobile sinks and section 3.4 describes the MMM and WSN delay tolerant characteristics.
- **Chapter 4:** In this chapter it is shown through Qualnet simulations that the MMM cannot be supported by routing protocols recommended for MANETs

such as: LAR, AODV, DSR, Bellman Ford and DYMO. This chapter starts by analysing three different WSN simulations environments such as the Network Simulator, Qualnet and MATLAB. For each of these simulation environments advantages and disadvantages are mentioned. In section 4.2 the five routing protocols are presented and in section 4.3 the results obtained from the Qualnet simulations are discussed. It is shown that, when the distance between the mobile sink and the WSN increases, which means that the D to WSN connectivity time decreases, the performances of the five routing protocols are poor in terms of total number of sent packets and average end to end delay. Poor performances are also obtained when the speed of the mobile sink increases.

- **Chapter 5:** In chapter five the PRP is described and the following discussions are made: when the use of PRP is needed; how D next neighbour ID is estimated; how the deconnectivity time between D and its next neighbour is estimated and what recovery measures PRP has for extreme cases.

- **Chapter 6:** In this chapter an analytical model for both PRP and LAR is presented for the throughput and energy. These models are discussed and compared and the following conclusions are stated: PRP will send more data packets when compared with LAR if more then two hops are between S and D; the energy spent on route requests packets by LAR is bigger than the energy spent on hello packets by PRP plus the energy to find a path from S to the next neighbour of D. In section 6.3 the impact of location errors on both PRP and LAR performances is discussed.

- **Chapter 7:** In chapter seven the results obtained from a MATLAB simulation in which D follows the MMM and LAR, AODV, DSR and PRP are used as routing protocols are presented. The chapter starts with a description of the used MATLAB simulator. It continues with a description of the simulation scenario and the results obtained. It is shown that when the D to WSN connectivity time varies from 10% to 95% PRP has the best results in terms of throughput, average end to end delay and energy. Also, PRP has better performances over LAR when the WSN size, density and D speed are varied.

- **Chapter 8:** This final chapter states the conclusion and future work together with the PRP constrains.

The papers published during the research period are listed in the next section.

1.6 Publications

The research papers generated during this study are listed as follows:

I. Journals:

1. I.G. Tudorache, and A.H. Kemp. “LAR and PRP Performance Analysis Subject to a Realistic Sink Movement”, submited to Ad Hoc Networks journal from Elsevier.

2. A. M. Popescu, I. G.Tudorache, A. H. Kemp, “Surveying Position Based Routing Protocols for Wireless Sensor and Ad-ho Networks” - published in International Journal of Communication Networks and Information Security, Kohat University of Science and Technology (KUST), Pakistan, vol 4, no. 1, 2012

II. Conference papers:

1. I.G. Tudorache, A.M. Popescu, and A.H. Kemp., “Improved mesh WSN support for a realistic mobility model”, in Wireless Communication Systems (ISWCS), 2010 7th International Symposium on, pages 340–344, 2010.

2. I.G. Tudorache, A.M. Popescu, and A.H. Kemp., “MANET routing protocols problem for the marginal mobility model”, in Microwave Conference (EuMC),

2011 41st European, pages 139–142, 2011.

3. Tudorache, I.G.; Rasool, I.; Kemp, A.H., "Indoor RSSI-based ranging consistency and error factors in wireless sensor networks", Telecommunications Forum (TELFOR), 2012 20th , vol., no., pp.607,610, 20-22 Nov. 2012.

4. Popescu, A.M.; Tudorache, G.I.; Kemp, A.H., "Performance study of node placement for geographic routing in WSNs", Communication Technologies Workshop (Swe-CTW), 2011 IEEE Swedish, pp.13-18, 19-21 Oct. 2011.

5. Rasool, I.; Tudorache, I.G.; Salman, N.; Kemp, A.H., "Range filtration algorithm for wireless sensor networks", Telecommunications Forum (TELFOR), 2012 20th , vol., no., pp.180,185, 20-22 Nov. 2012.

2 Sink mobility in WSNs

The current research analyses WSN scenarios which make use of mobile sink nodes. This chapter explains the assumption of use of a mobile destination at the periphery of a WSN. The mobility models used in WSN simulations are first presented in Section 2.1. Section 2.2 then presents a military application scenario for which the Marginal Mobility model (MMM) is defined. The chapter ends with a short discussion about the MMM characteristics.

The need for a mobile sink at the WSN periphery can be justified from different points of view, such as:

1. Application requirements - certain WSN scenarios (such as the military application presented in [8] and detailed in Figure 2.3) require data gathering in such a way. Military applications have very stringent requirements when it comes to real time data delivery. Battlefield deployed sensor nodes send the collected information (radiation measurements, movement parameters) to the battle station which is outside the war-zone. For this, a military vehicle (military robot or a mobile node like RacemoteZ [1]) needs to collect data from the battle field without entering the war zone. Traditional data gathering relies on a large number of densely deployed sensor nodes with a short radio range. The nodes form a well-connected end-to-end network and collaborate to collect and transmit the information. Mobility of nodes is often assumed necessary

in this process [1].

2. WSN energy saving - the sink movement maximizes network lifetime offering path diversity and preventing the case where the same routes are used excessively [44], [45]. The authors prove that if the destination moves at the network periphery, the overall WSN lifetime is improved by up to 500%.

3. No energy constrains for the sink - the battery of sensor nodes can be difficult or impossible to replace (due to remote or hostile environments) however, the mobile sink can have more energy available. Energy is one of the most restricted WSN resources, which makes economizing very important. Its efficient utilization is imperative. In [44] WSN lifetime is defined as the time duration until the first node exhausts its energy.

Different mobility models are used in the literature to simulate node movements in WSN. These are discussed further with an emphasis on the network simulation boundary imposed by each model.

2.1 WSN mobility models

A mobility model should realistically describe the behaviour of mobile nodes in a mobile ad-hoc network. The authors in [29] support the fact that mobility models directly impact the performance of routing protocols so using an unrealistic mobility model will lead to inaccurate performance measurements. In [46] the authors simulated multiple environments where mobility models are used to analyse the performance of different routing protocols. In [30] it is stated that past evaluations of multicast routing protocols have utilized a single, simple random way point model e.g. [31], [32], [33] and [34]. So the majority of existing mobility models for ad-hoc networks do not capture the variety of mobility patterns likely to be exhibited by ad-hoc applications.

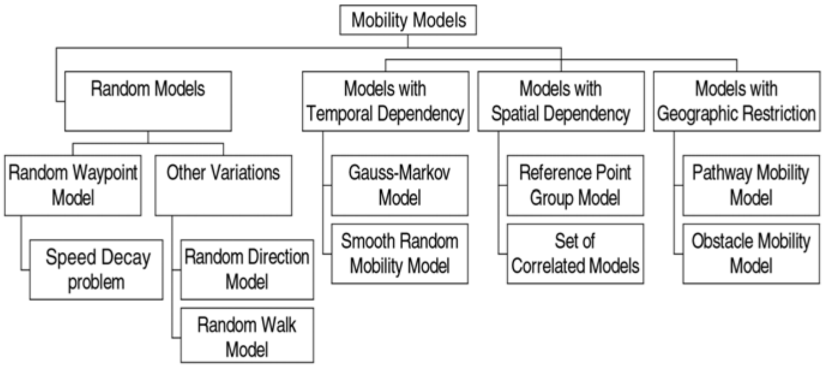

Figure 2.1: Mobility models classification in WSN [47]

In [47] the mobility models are grouped in four main categories (Figure 2.1) such as:

- **Random Models**: Random Waypoint model, Random Direction model and Random Walk model. The nodes move randomly without any restrictions where the destination, speed and direction are chosen randomly and independently of other nodes. Due to its simplicity and wide ability, the Random Waypoint model [48] became a benchmark model to evaluate the MANET routing protocols.
- **Models with Temporal Dependency**: Gauss-Markov model and Smooth Random Mobility model. The main characteristic of these models is that different parameters such as acceleration, velocity and rate of change of direction can depend on previous values. For example, the current velocity of a mobile node may depend on its previous velocity and this characteristic can be called the temporal dependency of velocity (Gauss-Markov Model). The memoryless nature of random models make them inadequate to capture this temporal dependency behaviour.
- **Models with Spatial Dependency**: Reference Point Group model and a Set

of Correlated models. Different parameters of a mobile node can be influenced by neighbouring nodes. For example, the velocity of neighbouring nodes can influence the mobility of other nodes. Since the velocities can be correlated in space, this characteristic can be called the spatial dependency of velocity.

- **Models with Geographic Restrictions**: Pathway Mobility model and Obstacle Mobility model. In the Random Waypoint model, the mobile nodes move freely and randomly anywhere in the simulation field. In some real life applications the node's movement is subject to the environment such as the motions of vehicles, which is bounded to the freeways. This type of mobility model can be considered a mobility model with geographic restriction.

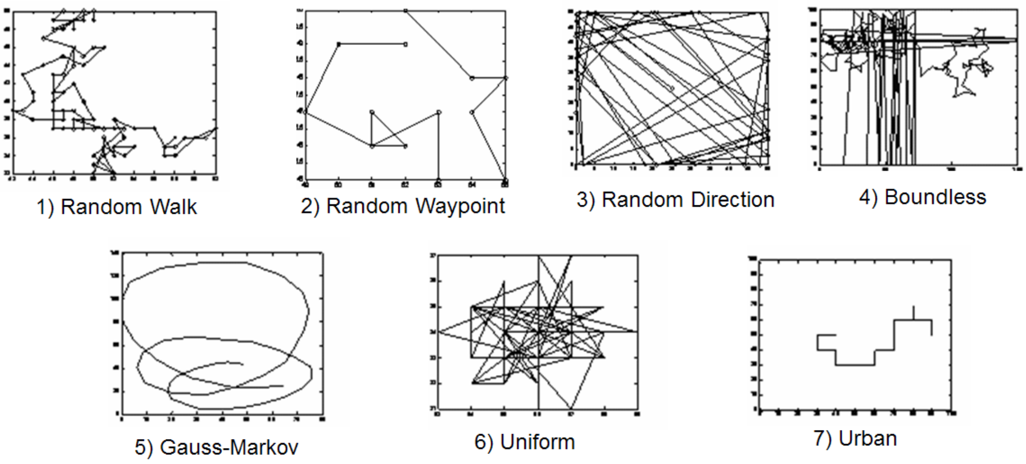

Figure 2.2: Mobility Models traveling patterns
[49]

The authors in [50] and [49] analyze seven mobility models in terms of traveling patterns of the mobile node. The mobile nodes do not exit the WSN coverage area for all of these seven mobility models. The mobility models which are used are the ones presented in Figure 2.2 and described as follows:

- Random Walk Mobility model: the mobile node mimics an erratic movement by randomly choosing its direction from the range of $(0,2\pi)$ and its speed from predefined ranges (v_0, v_1) independently of all previous values. When a mobile node reaches the simulation boundary it bounces back with an angle determined by its direction of movement.

- Random Waypoint Mobility model: this model includes pause times between changes in direction and/or speed. The speed that is chosen is uniformly distributed between a minimum and a maximum value. If the pause time is zero, then this model is similar to the Random Walk Mobility model.
- Random Direction Mobility model: a mobile node chooses a random direction and travels to the border of the simulated network area. When the border is reached, the mobile node pauses for a specified time after which it will chose a new angular direction and the process is repeated.
- A Bound Less Simulation Area Mobility model: a relationship between the previous direction of travel and velocity of a mobile node with its current direction of travel and velocity exits. A mobile node that reaches a side of the simulation area will continue traveling and reappear on the opposite side of the simulation area.
- Gauss Markov Mobility model: it was developed to adapt to different levels of randomness with the help of one tuning parameter. The speed and direction of the n^{th} instance is calculated based upon the value of the $(n-1)^{th}$ instance and a random variable. This model can eliminate the sudden stops and sharp turns of the Random Walk model.
- Uniform Mobility model: it collects good features of Random Walk, Random Waypoint and Random Direction Mobility models. For a node, the initial position is chosen in two steps: an initial path is chosen first after which a point on the path. The probability density of any chosen path is proportional to the length of the path.

- Urban Mobility Model: this model is able to mimic a real city where users who reach a boundary can freely pass through it.

The authors prove in [50] that from these seven models only Boundless Simulation Area and Urban Mobility models do not restrict the movement of nodes in the network simulation boundary (Figure 2.2). In the first one, when a mobile node reaches the simulation boundary it will continue to travel and reappear on the opposite side of the simulation area. In the Urban Mobility model users who reach a boundary can freely pass through it.

None of these mobility models consider the scenario where a sink will frequently exit the WSN during its peripheral movement. The mobility model needs to be carefully chosen when evaluating a routing protocol. The performance of the network is widely influence by mobility models, packet injection rates and speeds [51]. Because of this, if a certain application scenario is considered then the mobility scenario will be chosen to meet exactly the application requirements.

A WSN military application scenario is proposed next in which the sink is moving at the network periphery. Throughout its movement the sink will frequently enter and exit the WSN's coverage area. This type of movement is introduced as the Marginal Mobility model (MMM) and it will be defined as a new and realistic mobility model.

2.2 The Marginal Mobility model (MMM)

The military application scenario presented in Figure 2.3 is used to explain the MMM. Sensors are deployed in the battlefield area from which a mobile sink (node M) needs to collect data. Node S is the source which sends data packets to the mobile sink. It is consider that the WSN area (the battlefield) is a dangerous zone which needs to be avoided by the sink. Because of this the sink will move at the WSN

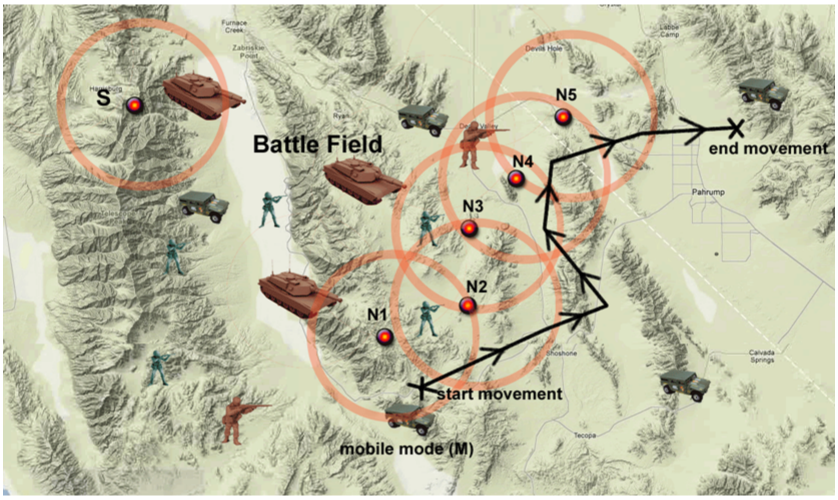

Figure 2.3: WSN military application where the sink follows the MMM

periphery throughout the coverage areas of nodes N1 to N5 which are peripheral nodes. It can be seen that, throughout its movement, the sink will enter and exit periodically the WSN coverage area because of its peripheral trajectory. This will mean that the sink will connect and disconnect frequently with the WSN. For this type of scenarios where the sink moves at the WSN periphery it can be said that the MMM is a realistic mobility model.

MMM is a model with geographic restrictions similarly to the Freeway Mobility model [52]. However, in the Freeway Mobility model the movement is restricted to a pathway in the simulation field (pathway 1 in Figure 2.4) where for the Marginal Mobility model the movement is restricted to a pathway which is at the WSN periphery (pathway 2 in Figure 2.4). MMM characteristics are analysed and presented next.

It can be assumed that the peripheral nodes coordinates are either known (from the simulator) or determined (this process is described in Section 5.1). Four vectors can be built: P_N, P_S, P_E, P_W which are the set of peripheral nodes from the north, south, east and west side of the network. Let N_1, with $N_1 \in P_N$ of coordinates (x_a, y_a) be

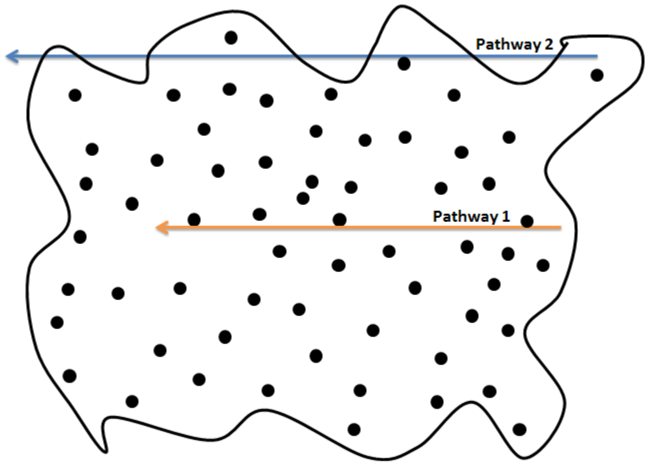

Figure 2.4: Pathway difference for the Freeway (pathway 1) and the Marginal Mobility Model (pathway 2)

a peripheral node. If the sink will move at the WSN northern periphery then the initial position of the sink can be (x_m, y_m) where $x_m = x_a + d$, $y_m = y_a + d$ and $d \leq r$ where r is the communication range of node N_1 and d is a random value. This condition will ensure that the sink starts its movement from a peripheral location (from a peripheral node's communication range) connected with the WSN. After the starting point of the movement has been established, the next step is to define the pathway of the node. For this two different options are proposed:

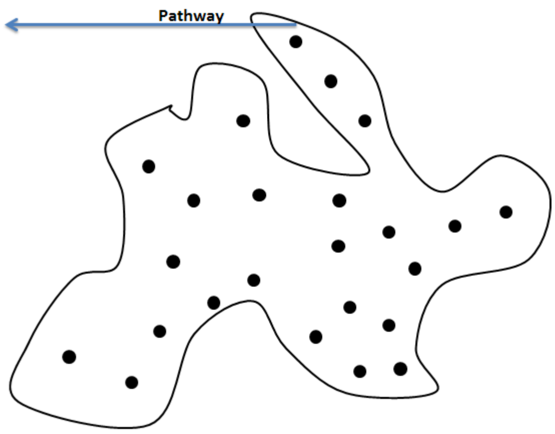

Figure 2.5: Extreme scenario for linear pathway

- Linear pathway - it is considered that the sink will move following a straight line from the starting point (x_m, y_m) with a constant or variable speed and acceleration. It is expected that the sink will enter and exit the WSN frequently during its peripheral movement (Pathway 2 in Section 5.1). However for some scenarios defining only the starting point for the sink will not be enough. The scenario presented in Figure 2.5 can appear when the sink is connected with the WSN only for a short period of time during its entire peripheral movement. For this type of scenarios a predefined pathway is preferred and this process is explained next.

- Predefined pathway - the sink is moving to predefined check points (C_1 to C_6) calculated based on the coordinates of the peripheral nodes. This scenario is presented in Figure 2.6 were the starting point C_1 is the one defined previously of coordinates (x_m, y_m). It can be seen that C_2, C_4 and C_6 are within WSN coverage area and that C_3 and C_5 are outside the WSN. This will enable the situation in which the sink will connect and disconnect with the WSN area frequently during its movement. The coordinates of the check points $C_i\,(x_{ci}, y_{ci})$ are calculated with:

$$x_{ci} = \begin{cases} \frac{x_{Ni}+x_{N(i-1)}}{n} \pm \alpha_1, & i\epsilon\,\{2,4,6\} \\ \frac{x_{Ni}+x_{N(i-1)}}{n} \pm \alpha_2, & i\epsilon\,\{3,5\} \end{cases}$$

$$y_{ci} = \begin{cases} \frac{y_{Ni}+y_{N(i-1)}}{n} + \beta_1, & i\epsilon\,\{2,4,6\} \\ \max\left(y_{Ni}, y_{N(i-1)}\right) + \beta_2, & i\epsilon\,\{3,5\} \end{cases}$$

where (x_{Ni}, y_{Ni}) are the coordinates of the peripheral node from the left and $\left(x_{N(i-1)}, y_{N(i-1)}\right)$ from the right of each check point C_i. It can be seen that for $i\epsilon\,\{3,5\}$ to have check points outside the WSN the following condition needs to be respected: $d(N_i, C_i) > r$. Different sink trajectories can be obtained by varying the values of n, α_1, α_2, β_1 and β_2 which will lead to different pathways. In Figure 2.6 a pathway obtained for $\alpha_1 = \alpha_2 = 0$, $n = 2$, $\beta_1 = 0$ and $\beta_2 = 2 * r$ is presented.

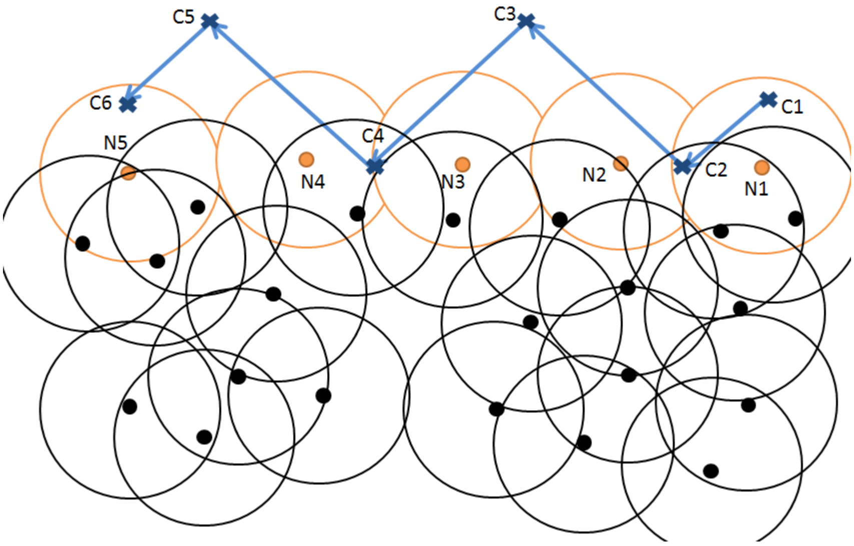

Figure 2.6: Predefined pathway for the MMM

The sink's speed for the MMM can be randomly chosen from predefined ranges (v_0, v_1) similar to the Random Walk Mobility model or it can be uniformly distributed between a minimum and a maximum value such as the Random Waypoint Mobility model.

MMM can be seen as a mobility model with geographic restrictions. The main contribution of this model is in defining a pathway throughout which a mobile sink connects and disconnects with the WSN frequently. Nodes can also follow the MMM if they move in the network throughout voids, i.e. areas without connectivity, but the number of voids needs to be significantly high to mimic this connected-disconnected behaviour. This type of movement will cause problems for routing protocols trying to send data to a sink which follows the MMM because the Random Waypoint model became a benchmark model to evaluate the MANET routing protocols.

The next chapter presents different routing protocols and location prediction schemes proposed for WSN scenarios with mobile sinks are presented. These will be analysed taking into account the military application scenario and the MMM model characteristics presented in this chapter.

3 Sink's mobility support in WSN

This chapter is divided into four sections. The first one presents a classification of routing protocols and specifications. The following three sections analyse and discuss current current state of the art routing protocols recommended for Mobile Ad-hoc Networks (MANET) (Section 3.2) location prediction schemes (Section 3.3) and Delay Tolerant Mobile Sensor Networks (DTMSNs) (Section 3.4) proposed for WSN scenarios where sinks are mobile and have MMM characteristics.

Routing in WSN is very challenging due to several characteristics that distinguish them from wireless ad-hoc networks such as [53], [36], [54]:

- Addressing scheme - classical IP based protocols cannot be used in WSN because an addressing scheme for a large number of sensor nodes is not a viable solution.
- Data flow - the majority of WSN applications require data transmission from multiple sources to a particular sink.
- Data redundancy - in some applications (such as monitoring applications) multiple sensors may generate the same data which leads to a significant data redundancy. In order to save power this data redundancy needs to be considered by the routing protocols.
- Thorough resource management - this needs to take into account constraints

such as: transmission power, on-board energy, processing capacity and storage.

Routing protocols need to consider the characteristics of sensor nodes, application and architecture requirements. WSN routing protocols classifications and specifications are presented further.

3.1 Routing protocols classification and specifications

A routing protocol survey is made in [36] from which the following routing classification can be summarized:

- Depending on what drives the routing: node centric, data centric, Geo-centric, QoS-centric.
- Depending on demand and start-up point of route: source or destination initiated.
- Depending on memorization: reaction based and positioned based. The reaction based category can be further grouped as:
  - proactive routing (table driven), nodes memorize tables with global position information and paths ahead of events.
  - reactive routing (demand driven), nodes memorize tables with global position information and creates routes on demand.
  - hybrid routing, combines proactive and reactive routing and uses flooding.
- Other routing categories:
  - topological classification, such as: flat based routing (all nodes are equal and are treated accordingly), hierarchical based routing (nodes are grouped in clusters and have different power levels).

- situation aware classification, such as: adaptive and static routing. For adaptive routing routes are chosen according to the traffic and power consumption. In static routing these are not considered when routes are selected.
- depending on the position and number of destinations the routing can be classified as multi-cast and geo-cast routing.

The routing protocol performance is closely related to the architectural model which is directly influenced by the application requirements. Architectural issues which influence the routing protocol design are explained further [53]:

1. Node deployment - routing protocol performance is affected by the topological deployment of sensor nodes. The deployment can be deterministic (the data is routed using pre-determined paths) or self-organizing where the sensors are scattered randomly and in which the sink position will drastically influence the overall WSN performance and energy efficiency.

2. Energy consideration - the energy is the most important resource of a WSN. Depending on the chosen metric of neighbour selection for the path formation between sender and sink, significant overhead for topology management and MAC can be introduced in the WSN. This will lead to more energy consumption so, based on the application requirements, the routing protocol metric needs to be carefully considered.

3. Data delivery models - this can be continuous, event driven, query-driven and hybrid [55] which will influence the routing in terms of energy efficiency and route stability. Depending on the application requirements a different model is needed. An example can be an application where data is continuously delivered to the sink. For this scenario a hierarchical routing protocol is recommended

to reduce the traffic and save power because redundant data can be aggregated [56].

4. Node capabilities - different types of sensors can be used in a WSN (such as: sensors for monitoring temperature, pressure, humidity, etc.) which lead to data reading at different rates. This can also mean constraints related to different QoS and data delivery models, therefore the data routing for these environments is more challenging.

5. Data aggregation/fusion - the authors in [57] define data aggregation as the combination of data from different sources by using functions such as suppression (eliminating duplicates), min, max and average. Stating that computation is less energy consuming than communication [58], data aggregating would lead to energy savings.

6. Network dynamics - in general it is assumed that sensor networks are static, besides few setups that use mobile sensors [59]. However, some authors specify the need of mobile nodes in wireless sensor networks [1]. Sending data packets from or to moving nodes is more challenging since route stability becomes an important optimization factor, in addition to energy, bandwidth etc.

Routing performances are also influenced by important design parameters as follows [36]:

1. Loop Free - the same data can circulate in the network many times using the same paths without reaching the destination leading to low packet delivery ratio (PDR) and high energy consumption. This needs to be avoided.

2. Distributed Operation - networks can operate in a centralized, decentralized or distributed manner. For localized algorithms a node will use information about its position, its neighbours and destination to make forwarding decisions. In

non-localized algorithms global or zonal information is used instead [60] which means that routing tables are maintained at each node. This will increase the overhead, energy expenditure and will make these algorithms less scalable thus localized algorithms are preferred.

3. Path Strategy - single path, multipath or a combination of these two strategies can be used to find a path between the source and destination. To preserve the network resources an algorithm with a single path strategy [61] is preferred.

4. Packet Forwarding - The routing decision is made according to a forwarding strategy which does not guarantee that a packet will reach the destination [62]. There are three main forwarding strategies which can be used: greedy [63], [64], flooding [37], [65] and hierarchical [66]. Greedy forwarding will choose to forward packets to nodes that are always progressively closer to the destination. Flooding will send the packets to all the neighbours and flood the entire WSN. Hierarchical routing uses a hierarchical addressing scheme to choose the next hop. A forwarding strategy can use different metrics such as: hop count, geographic distance, progress to destination, direction, power, cost, delay or a combination of these [67], [68], [69], [70]. If the message has reached a node which has no closer neighbours to the destination (a void or hole), a recovery procedure is necessary making the forwarding method a hybrid. Recovery from such a concave node can be done through flooding [71], [72], [73], [9] or perimeter (face) forwarding [62], [74], [75], [76].

5. Path Selection Metrics - the hop count, power and cost metric [60] are the most used metrics however many more are presented in [77]. The path selection metric needs to consider the routing goal which can be power efficiency, real time routing, throughput maximisation, etc.

6. Memory (state) - routing algorithms can have memory requirements for nodes

or not (stateless algorithm). A minimum requirement is the one in which nodes need to retain information about their position, neighbours and destination positions. Additional information can be added such as: speed, links cost, nodes communication range, energy level, cryptographic keys, etc. It is desirable to avoid the situation where a large amount of information needs to be gathered from the WSN and maintained at each node because this can create high traffic, queues, congestion, overhead, latency, energy expenditure [60].

7. Guaranteed message delivery - messages delivery from a source to a destination for storage or further processing is guaranteed by routing algorithms [61]. The data delivery between two neighbouring nodes is guaranteed either at a routing level or at the MAC level. As a consequence certain protocols choose to simulate the MAC layer when the delivery performance is analysed.

8. Scalability - network performance does not need to be affected when the WSN size is increasing by more nodes joining in. The performance of a routing protocol is influenced by the metric for which it was optimized such as obtaining the best throughput, the lowest energy consumption, the smallest delays, etc.. Scalability is not measured in a particular way so stating that an algorithm is, or is not, scalable is rather subjective.

9. Overhead - this can be classified as packet overhead or processing overhead. Packet overhead leads to traffic overhead which means that a high number of packets (data packets, control packets, acknowledge packets) will lead to an increase in bandwidth consumption and data processing requirements. The routing protocol design is very important because this can lead to a combination of unnecessary resource expenditure, such as computation time and energy, memory and bandwidth.

10. Adaptive to mobility - depending on the application requirements a routing protocol may need to support mobile nodes. Node mobility will induce network topology changes which can lead to an increase in traffic, overhead and energy consumption [78].

11. Additional data - a routing protocol needs to be designed to meet the purpose of a certain application so different requirements need to be addressed such as network type, network recommendations (size, density, mobility) and transmission type.

Different routing protocols proposed to support sink mobility in WSN are discussed in the next section. It will be analysed how these can cope with a mobile sink which follows the MMM.

3.2 Routing protocols used for mobile sink scenarios

To cope with the sink's mobility, routing protocols choose different schemes such as: use location updates through beacons (LAR [37], AODV [39], MRLG [79], CTP [80], Direct Difussion [81] , Declarative Routing Protocol [82], GRAB [83], MobiRoute [84]), keep tables (DREAM [85], DSR [38], Termit-hill [86]), use a forwarding area shift (IGF [68]), use the perimeter forwarding (GPSR [74]), use partial flooding (TTDD [9]), or a long time for pauses (MobiRoute [84]). These types of routing protocols are discussed next. Another aspect which is studied herewith is how these protocols support the MMM.

IGF (Implicit Geographic Forwarding) [68] is a state-free location based routing protocol designed to perform optimally in dynamic scenarios. The authors state that "routing to locations" takes place and not to "individual sensing devices". Figure 3.1 depicts an example of a topology where: source node (S) is transmitting a message towards the destination (D); r is the communication range of S; node R and A represent the chosen next hop; N represents a node within communication range of S that is not a candidate node. IGF uses the increased distance toward the destination and energy remaining metrics into the route selection process. Candidate nodes are the ones within a 30-degree angle of the line connecting S and D. It can be seen that IGF does not establish a path first, it sends data trying to find a path. When D moves at the WSN periphery following the MMM, the line connecting S and D is continuously changing. Because of this S will fail to deliver data packets to D. The data packets will be dropped once these reach the peripheral nodes and D is not connected with the WSN. At this point S restarts the data transmission using the previous known coordinate of D.

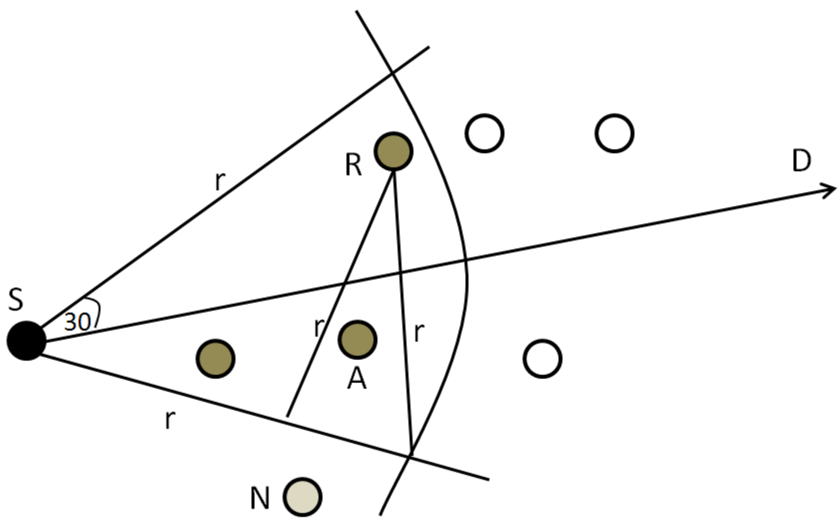

Figure 3.1: IGF Forwarding area for source S [68]

In GPSR (Greedy Perimeter Stateless Routing for Wireless Networks) [74] the positions of routers and destination are used for packet forwarding decision. GPSR uses local topology information to find new routes quickly when the WSN topology changes. It uses greedy forwarding and perimeter forwarding to select the next best candidate to route the data packet, see Figure 3.2. When D follows the MMM and breaks the communication with the WSN, its neighbour will start a perimeter forwarding procedure. GPSR will remain stuck with this procedure if *D* moves continuously and throughout its movement it will connect and disconnect with the WSN frequently.

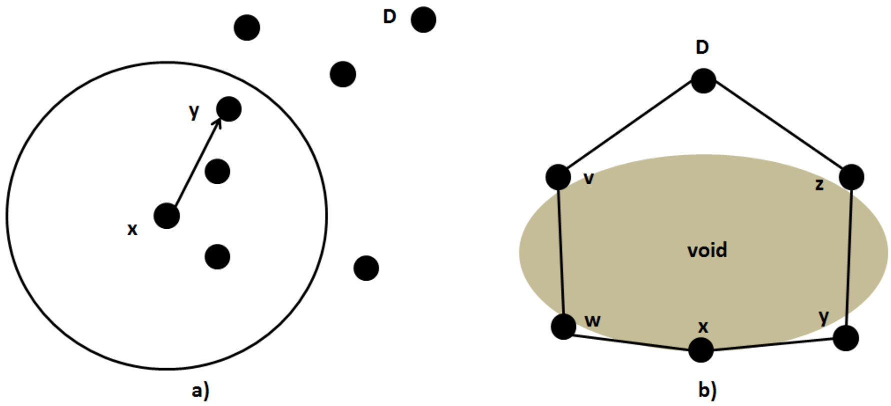

Figure 3.2: a) Greedy forwarding example, y is x's closest neighbour to *D*; b) Node x's void with respect to destination *D*
[74]

MRLG (Mobile-sink Routing for Large Grids) [79] is a routing protocol which supports highly mobile sinks that move freely throughout the sensed area. The sink broadcasts RREQ packets during its movement to inform all the WSN nodes about its position. The rate at which these RREQ packets are sent can be changed according to its velocity. When a link to reach a certain node is broken, its routing entry is removed. MRLG will select the entries with fewest hops to reach the sink from the list of downhill and peer nodes. If no next hop neighbours are found, the sensor nodes will not send any data until the sink refreshes the routes. When the sink follows the MMM and the link between it and the rest of the WSN breaks, the sink tries to refresh the entire route. However this process will not be possible when the sink moves outside the WSN area. Data packets reach the sink when it reconnects with the WSN, only after the routes are refreshed. The sink will not receive any data packets if its connectivity time with the WSN is less then the time needed by the sink to refresh the routes.

TTDD (Two-Tier Data Dissemination) [9] is designed to support multiple mobile sinks. A sink will broadcast in the entire WSN information about its location. Multiple mobile sinks will lead to an increase in WSN energy consumption and packets collision. TTDD uses grid structures constructed by each data source. Queries are flooded within a local cell only (the cell in which the sink is) to make possible for the sink to receive data continuously through its movement. A WSN scenario in which TTDD is used is presented in Figure 3.3 where mobile sinks (soldiers) need to receive data continuously. When the sink follows the MMM and breaks the communication with the WSN the cell corresponding to that sink is flooded with queries. This process will continue until the sink rejoins the WSN which leads to high energy consumption. Also, the sink will rejoin the WSN during its movement, but it will use a different cell. This means that the grid structure needs to be periodically refreshed by the source which leads to more energy consumption.

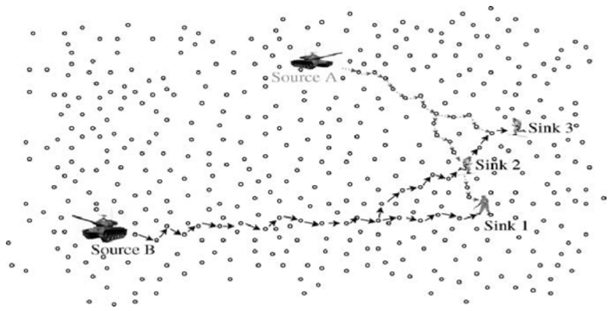

Figure 3.3: WSN scenario in which TTDD is used
[9]

MobiRoute [84] describes a scenario in which the sink is mobile and has high energy reserves and all the other nodes are fixed with low energy reserves. The data delivery delay induced by the sink's high speed is considered to be tolerable because the sink will pause its movement frequently at different anchor points. For the routing overhead induced by the sink's mobility to become negligible, it is required for the pause periods to be much longer than the moving periods. In MobiRoute beacon messages are periodically sent by the sink. If a node misses the next beacon after a certain time this will detect a link breakage and a new parent is chosen (which is taken care by MintRoute [87]). The rate at which beacons are sent needs to be carefully considered to obtain a low energy consumption and not flood the WSN. The data delivery delay, routing overhead and energy spent by the WSN will increase if the sink will follow the MMM with no pause time throughout its movement. Also, no data packets will reach the sink if it's connectivity time with the WSN is smaller than the time needed to establish a path.

Termit-Hill [86] is a routing algorithm that supports mobile sinks where data packets are sent towards a strong pheromone path and the selection of next hop is always randomly decided. Termite-hill uses routing tables and discovers routes only when these are required. If no route is found in the tables then the node broadcasts the request further. It can be said that the process to find a route towards a destination which follows the MMM is time and energy consuming. This will not be successful if the sink is connected with the WSN for short periods of time throughout its movement. Also the tables needs to be refreshed periodically because the sink changes its position which increases the energy consumption.

SEAD (Scalable Energy-efficient Asynchronous Dissemination protocol) [88] uses a dissemination tree to send data to mobile sinks with the purpose to minimize the energy consumption through the following steps: 1) it builds a dissemination tree; 2) it disseminates the data; 3) it maintains links (paths) to mobile sinks. To build an optimal dissemination tree, stationary sensor nodes (terminals) will take the place of the mobile sinks. To minimise the energy SEAD will select paths to these stationary terminals. The forwarding delay will increase when the sinks will move away from these terminals so there is a trade-off between minimizing this delay and reducing the energy spent on reconfiguring the tree. When the sinks follows the MMM data packets that reach a sink's terminal are dropped and a tree reconfiguration procedure is required. This process will take place each time sink exits the WSN coverage area which makes this protocol unsuitable for this type of mobility model.

Direct Diffusion [81] uses multiple gradient paths to send data from source to sink where one or a small number of these paths are reinforced by the WSN. A straightforward application of reinforcement rules will cause all nodes downstream of the lossy link to also initiate reinforcement procedures. This procedure will make sure that a good path is found, but it will also increase the energy consumption. A reinforcement procedure will start if the sink follows the MMM and breaks the communication with the WSN which leads to high energy consumption. In addition to this, there may not be sufficient time to find a path to the sink if the sink reconnects with the WSN for short periods of time. Consequently, this protocol is not suitable for the MMM model either.

DREAM [85] uses routing tables and a probabilistic method for selecting the direction in which a given node may be found. Routing information about the slower moving nodes needs to be updated less frequently when compared to highly mobile nodes. It considers a scenario in which mobile nodes continuously move in the WSN area. So, a sink which follows the MMM has high chances not be in the WSN coverage area when data packets reach the WSN periphery. So it can be concluded that DREAM will not support this type of mobility model.

MobiCluster[89] is proposed for a scenario in which a mobile node moves within a periodic schedule following a fixed peripheral trajectory. Once the mobile sink joins the WSN can start to receive data from a nearby cluster node. By using cluster nodes MobiCluster will save the peripheral nodes from data transmission. This enables a balanced energy consumption across the WSN. For a scenario in which real time communication is needed (the data needs to reach the sink as soon as it is generated), the sink moves at the WSN without following a fixed trajectory and a periodic schedule is not discussed. So the MMM characteristics are not supported by this routing protocol either.

Different location prediction schemes are discussed in the next section. The study is extended to analyse if the mobility models used for the simulations consider the MMM characteristics.

3.3 Location prediction schemes used for mobile sinks

A network-assisted sink navigation scheme is presented in [90] to analyse the stability and performance trade-offs of WSN with mobile sinks. The investigation also analyses the amount of multi-hop routing that can be used in stable and efficient data collection. The impact of multi-hop routing on energy and data collection delay performance is studied. It is stated that the two most important design issues are represented by the data collection delay and network lifetime. The best delay can be achieved for static sinks but mobile sinks increase the WSN lifetime. The trajectory followed by the sink should be designed based on a trade-off between the data delivery delay and lifetime. However the authors do not consider the MMM scenario where the sink connects and disconnects from the WSN frequently during its movement.

[91] studies the effect of inaccurate location information caused by mobility under different scenarios and mobility models. A scenario is proposed where GPSR uses the perimeter mode forwarding to cope with a local maximum problem. This scenario is defined by the authors as the LOOP problem. To solve this a destination location

prediction scheme (DLP) is proposed. In DLP, a node checks to see if the source is in its neighbour list before it makes a packet forwarding decision based on the location information of the destination. If it is found, then the data is sent to it without any other calculation. However, the situation where the sink joins and leaves the WSN frequently is not considered such with the MMM model. For this type of scenario DLP will not solve the local maximum problem where GPSR uses the perimeter forwarding.

In [92] the authors also study coordination and communication problems in a Wireless Sensor and Actor Networks (WSANs) with mobile actors is studied. In this network, sensors will detect events and send data using geographic routing to mobile actors. Location updating and location prediction is used to handle the mobility of actors with minimum energy consumption for the sensors. Actors that move whilst following predictable trajectories will need to update their position less frequently than actors that follow temporally uncorrelated trajectories. Sensors will estimate the location of their neighbouring actors with the help of a location prediction scheme which will be influenced by the accuracy of the mobility model and the efficiency of the prediction algorithm. The used mobility model does not consider the scenario where a mobile actor can leave and join the WSAN as frequently as with the MMM. Because of this the location prediction scheme proposed here will not support a mobile node which follows the MMM.

In [93] a location-delay prediction scheme, based on a location-resource update protocol, which assists a QoS routing protocol is presented. An update protocol is used which has two type of updates: a type 1 update which is generated periodically and a type 2 update which is generated when there is a considerable change in the node's velocity or direction of movement. These updates will help a QoS routing protocol to predict the location of a mobile sink at a certain period of time. However, the scenario in which the mobile sink will connect and disconnect frequently with the WSN throughout its movement is not analysed.

In [94] aspects like a mobile node losing its connectivity due to the issues caused by joining or leaving sensor nodes communication area within a WSN are discussed. When a mobile node exists the communication range of its neighbours, it will start a re-association procedure with a new coordinator which is in range. However, due to mobility, the time for the re-association procedure might be to small. To overcome this problem, a kernel moving average scheme is designed based on the quality indicator prediction in the media access layer. The proposed scheme can reduce the non-connectivity time of mobile sensor node through early performance of the re-association procedure. This procedure is similar to the hand-off procedure used in the cellular network. However the scenario in which the mobile node might stay disconnected with the entire WSN for a certain period of time is not discussed, hence the MMM model will not be supported.

In [95] a predictable mobility based communication paradigm that enables data to directly route from a source to a moving sink by exploiting prediction of sink trajectories is discussed. The mobile sink will need to move to a collection point to receive the data generated by the source. So, the scenario in which the sink needs to receive information fast is not considered. Also, the sink moves all the time withing WSN coverage area so the MMM scenario is not discussed. It can be concluded that this approach is not suitable for a scenario where the sink follows the MMM and it needs to receive fast information from the WSN.

In [96] a method is presented for efficiently and reliably routing data packets from a static information source to a mobile sink through a multi-hop wireless sensor network. The mobile sink estimates and tracks its location, speed and acceleration with a Kalman filter. The source predicts the location of the mobile sink. The prediction is updated by receiving messages from the mobile sink, containing its current location, speed and acceleration. These messages are not periodically sent, but only if the Euclidean norm of the error between the predicted state and the state estimated by the Kalman filter exceeds a pre-defined threshold. However when the mobile sink is not in WSN coverage area these updates are not sent. This will mean that the MMM characteristics are not supported.

In [97] it is examined how mobility prediction is use to anticipate topology changes and perform rerouting prior to route breaks. The link and route expiration time are used in this process. It is assumed that as soon as a link between source and destination breaks, a new link will be available. This means that the source is all the time connected with the WSN during its movement. Because of this assumption it can be said that MMM will not be supported by this mobility prediction scheme.

It can be concluded that the above location prediction schemes do not support the MMM because they do not take into account the scenario in which the sink throughout its movement will connect and disconnect with the WSN frequently. Because delay tolerant networks (DTNs) embrace the concept of occasionally-connected networks that may suffer from frequent partitions [43] these are discussed next.

3.4 MMM and WSN delay tolerant characteristics

When the sink moves at the network periphery following the MMM, there are two connectivity situations to consider: one regarding the communication of the nodes amongst themselves and one regarding the communication between all the nodes and the sink. The sink can become isolated and lose contact with the rest of the nodes. This circumstance results in a WSN with the characteristics of Delay Tolerant Mobile Sensor Networks (DTMSN), namely: intermittent connectivity, long or variable delays, asymmetric data rates and high error rates [43], [98], [99], [100], [101]. It is difficult to find a well-connected end-to-end path for all the sensors to transmit data to the sink when it follows the MMM because it will enter and exit the WSN coverage area frequently. This case is most often met in DTMSN networks which embrace the concept of occasionally-connected networks that may suffer from frequent partitions [43].

The assumption that is very difficult to form and maintain a path between sender and sink is common in DTMSN. This happens because the sink will often exit the WSN coverage area or it will move throughout voids within the network. Routing protocols mainly replicate the data and flood the network to increase the chances for it to reach the sink. Different improvements to this method are proposed with the purpose to reduce the energy expenditure and increase the overall throughput such as:

- A history based approach is presented in [102] (ZebraNet) and [103] (PROPHET) to reduce the number of duplicates packets from the network.
- A direct contact approach (DataMule [104]) were MULEs collect data from the sensors only when these are in close range after which the data is delivered to wired access points.
- In MPAD [105], Vector Routing [106] and MTAD [107] the messages are replicated (epidemic routing) and sent to nodes with a higher probability of meeting the sink.
- In [108] it is considered that the sink traveling time between stops is negligible.
- The movement of the mobile sink is controlled in [109] to collect static sensors reading.
- When and where to send data based on the delivery probability is presented

in [110].

- [111] prescribes a delay tolerance level for the data to reach the mobile sink.

In DTMSNs, the choice of an appropriate routing algorithm is essential because the successful establishment of end-to-end paths between the source and sink is not guaranteed. Moreover, typical MANET routing protocols may not support such characteristics because they can forward the data only when a path between sender and sink exists [43]. In a DTMSN delays and faults are tolerable. Since it is difficult to form a well-connected mesh network, the data delivery protocols for conventional sensor networks relying on end-to-end connections simply fail to achieve a satisfactory throughput and a tolerable delay [35], [43], [98], [112], [113], [114]. [101] surveys routing protocols proposed for DTMSNs and states that only high latency applications are used in such networks.

However, for the proposed military scenario where the sink follows the MMM from Section 2.2, using a DTMSN routing protocol is not a solution even if the MMM will add DTMSN characteristics to the WSN. The reason is that DTMSN routing protocols require from the application itself the DTN characteristics to set up the routing protocol. For our proposed scenario, the DTN characteristics are not known nor are they specified by the application, but by the WSN scenario. In our case we have a delay intolerant application (DIN) where the information needs to reach fast to the sink.

It was discussed in this chapter that none of the MANET routing protocols, location prediction schemes or DTMSNs routing protocols proposed to support mobile sinks scenarios can be used for the military application presented in Section 2.2 where the mobile sink follows the MMM (a DIN application where the WSN scenario has DTN characteristics). It can be said that due to MMM characteristics, routing protocols proposed to support mobile sinks in WSN will face the following challenges:

- Path formation and maintenance - it will be difficult to find a path to a mobile sink which will connect and disconnect with the WSN frequently throughout its movement. This might not even be possible if the connected time intervals are smaller than the time needed to find a path between sender and sink (the distance between the source and the sink is high).
- Keeping a small energy consumption figure - to cope with mobile sinks routing protocols choose to broadcast hello packets in the entire WSN to keep the sink locations up to date for all nodes or to replicate the data in the network and send it to all nodes. This process will consume a high amount of energy.
- Obtaining a satisfactory throughput - data packets will be dropped by sinks neighbours when this is not connected with the WSN. It is difficult to reduce the number of dropped packets if there is not knowledge about the time and place the sink will connect and disconnect with the WSN.
- Keeping a low end to end delay - it will be difficult to send fast data packets to a sink which follows the MMM for a real time communication scenario not knowing sink's exact position at a certain time.

In the next chapter three WSN simulators are analysed and compared. It will be shown through Qualnet simulations that five routing protocols do not support the MMM.

4 Proving that MMM cannot be supported by existing routing protocols

This chapter presents, through Qualnet simulations, that five MANETs routing protocols do not support a mobile sink which follows the MMM. In Section 4.1 three WSN's simulation environments are discussed and compared. The routing protocols used to send the data packets from the source to the mobile sink are presented and discussed in Section 4.2. These routing protocols were chosen because all of them are performant routing protocols recommended for MANETs and are available in the Qualnet library. The simulation topology used for the Qualnet simulations is presented in Section 4.3 together with the obtained results when the distance between the mobile sink and WSN is varied. Also simulations are made when different MAC protocols are used. Conclusions are presented at the end of the chapter.

4.1 WSN simulation environments

Three WSN simulators are presented with both advantages and disadvantages. The simulation process is a very important step in WSN design. The simulation process is chosen when:

- The WSN designer wishes to test different network performances.
- A new algorithm needs to be tested like a routing protocol, MAC protocol, etc.
- The interaction between different network layers needs to be observed.
- To predict different problems that can appear later at a WSN implementation stage, problems that can at this point lead to waste of money, time or hardware equipment.
- Different network parameters need to be tuned or calculated for different case scenarios before switching on a real WSN platform.

4.1.1 Network Simulator 2 (NS2)

The Network Simulator 2 (NS2) [115] is a powerful tool used for WSNs simulations. Some advantages and disadvantages are discussed as follows:

1. Advantages:
    - It is free, so there are many discussions and recommendations on how to use it.
    - You can run built in examples quick and obtain detailed results.
    - It is developing fast and new versions appear on a regularly basis. So it becomes more stable.

- The simulations are running fast.
- Obtain good trace files.

2. Disadvantage:

- It uses two programming languages, TCL and C++.
- Poor graphic visualization of scenarios/simulations which are limited to static scenarios.
- You cannot make quick modifications to your scenario (i.e. drag and drop).
- For adding new patches you need the right version for which the patch was developed.
- It is developing fast so it is harder to use the available code source on the web.
- When developing a new routing protocol you need to define functions in C++ and use them in the TCL script, so there are not pre-defined functions.
- Difficult to install on Windows, so you need Linux (or Unix) basic knowledge and two operating systems installed on your computer.

4.1.2 Qualnet simulator

QualNet Developer [116] provides a comprehensive environment for designing network protocols, creating and visualizing network scenarios under use-specified conditions, and analyzing their performance. QualNet Developer is composed of the following tools:

- QualNet Scenario Designer — A graphical scenario design tool.

- QualNet Animator — A graphical scenario animation tool.
- QualNet 3D Visualizer — A 3-D scenario animation tool.
- QualNet Analyzer — A graphical statistical analyzing tool.
- QualNet Packet Tracer — A graphical packet tracing tool.
- QualNet Command Line Interface — Command line access to the simulator

QualNet Developer enables users to:

- Design new protocol models.
- Optimize new and existing models.
- Design large wired and wireless networks using SNT-provided (Scalable Network Technologies) or user-designed models.
- Analyze the performance of networks and perform "what-if" analysis to optimize them.
- Perform cross-layer optimization of wireless network stacks.

The user can modify the: Application Layer, Transport Layer, Network Layer, Mac Layer, Physical Layer, Communication Medium, Node Mobility. The key features of QualNet Developer which enable the creation of a virtual network environment (VNE) are:

- Speed - Can support real-time speed to enable software-in-the-loop, network emulation, and hardware-in-the-loop modeling. Faster speed enables model developers and network designers to run multiple "what-if" analysis by varying model, network, and traffic parameters in a short time.
- Scalability - Supports thousands of nodes by taking advantage of the latest software, hardware and parallel computing techniques. The base QualNet Developer product can run multiple threads simultaneously to benefit from

the latest multi-core processors. Advanced versions of QualNet Developer can run on cluster, multi-core, and mulch-processor systems to model large networks with high fidelity.

- Model Fidelity - Uses highly detailed standards-based implementation of protocol models. It also includes advanced models for the wireless environment to enable more accurate modeling of real-world networks.
- Portability - Its library of models run on a vast array of platforms, including Linux, Solaris, Windows, and Mac OS operating systems, distributed and cluster parallel architectures, and both 32- and 64-bit computing platforms. Users can develop a protocol model or design a network in QualNet Developer on their desktop or laptop computer and then transfer and run it on a powerful mulch-processor Linux server to perform capacity, performance, and scalability analysis.
- Extensibility - Can connect to other hardware and software applications and third party visualization software, greatly enhancing the value of the network model.
- Compiling - In order to modify the QualNet source code, or add customized models, a recompilation of the main executable file is needed. The QualNet installation guide provides instructions for compiling QualNet on Windows and UNIX (Linux, Solaris, and Mac OS) platforms. Instructions for activating add-ons can also be found in the installation guide.

The following advantages are specific to QualNet:

- Easy to start, a first simulation can be run in less then 1h.
- Easy to modify an existing scenario by changing routing protocols, MAC protocols, mobility models, etc.

- Easy to visualize and analyze the obtained simulation results.
- Run multiple simulations in batch mode.
- Specify range of values for each variable and run all possible combinations.
- Compare multiple sets of data on the same graphic.
- Modify the existing protocols and create new protocols with the help of pre-defined functions in an C++ environment (Microsoft Visual Studio)
- Graphical visualization.
- Available documentation.

Also there are some disadvantages like:

- Not a free software so it is not used frequently.
- It will be hard to share your problems with others.

4.1.3 MATLAB simulator

MATLAB [117] is a powerful mathematical simulation tool. For WSNs it is used to simulate different mathematical models used in routing, MAC protocols or other layers. Some advantages and disadvantages are presented further.

Advantages:

- Good to test an idea for a certain layer (ex: routing, localization, etc.) .
- Good to analyze the data obtained from real-time (hardware) simulations, 80% are using it.
- Good to develop mathematical models.
- Fast simulations.
- Mathematical based language simulation which is easy to use.

- Well known, well used, well developed software.

Disadvantages:

- No WSN library, no stack layers so it is difficult to simulate an entire WSN topology.
- A mathematical model needs to be built to test ideas so advance mathematical skills are required.
- It is not a free software.

It can be drawn the conclusion that choosing the right simulation environment can be a very important task because of different advantages and disadvantages that each simulation tool has. For simulating mathematical models one can use MATLAB. For simulating the entire layers of a wireless sensor network NS or Qualnet can be considered. Qualnet is used in this chapter (Section 4.3) to prove that five MANETs routing protocols do not support a sink which follows the MMM. Why these five routing protocols were chosen for the simulation is discussed next.

4.2 Routing protocols used for the Qualnet simulation

It was desired to observe the effects of the MMM on routing protocols which use different strategies to find and maintain a path between the source (S) and destination (D). The following five routing protocols with different characteristics were chosen:

- LAR - does not keep any tables and uses a route request zone to reduce the energy consumption. A path from S to D is searched when needed which means that this is a reactive routing protocol.

- DSR - is also a reactive routing protocol where S saves in cache different routes to D. When a route is broken the next one is tested. When all the routes are broken a path discovery process starts and route request packets are sent in the entire WSN.
- AODV - is also a reactive routing protocol and it uses traditional routing tables with only one entry per destination. It uses sequence numbers to avoid routing loops and keep updated the routes. Hello messages are used to maintain paths.
- DYMO - enables reactive, multihop unicast routing between participating routers and uses a route discovery and a route maintenance procedure. The links over which traffic is flowing are monitored to keep the routes fresh and react to topology changes.
- Bellman Ford - is a proactive algorithm where each node maintains tables with different cost functions to all the other nodes from the network.

Many authors used LAR, AODV and DSR to compare their performance with new proposed routing protocols designed to support sink mobility in WSN or for different mobile WSN scenarios analysis such as:

- In [118], [119], [120], and [121] AODV is used to compare the performance of different mobility models in WSN, to evaluate different mobility conditions or to analyse the impact of mobile sinks.
- The Spatial Vector Routing Protocol [122], the Energy Aware Geographic Routing Protocol (EAGPR) [123], SPEED [124], GPSR [74], Termite-Hill [86], IGF [68] and DREAM [85] are compared with LAR, AODV or DSR.

Bellman Ford and DYMO are two protocols recommended to support mobility in WSN and were also chosen because of their availability in Qualnet. A short analysis of these five routing protocols functionality is presented further.

LAR [37] uses a request zone that is rectangular in shape (as depicted in Figure 4.1) to limit the area in which a path from the source (S) to the destination (D) is searched. It is assumed that S knows the coordinates (X_d, Y_d) and speed of D at a certain time t_0. At time t_1 node S initiates a new route discovery for destination D and it defines the expectation zone for D to be the circle of radius $R = v * (t_1 - t_0)$ centered at location (X_d, Y_d). When a node receives a route request, it discards the request if the node is not within the rectangle specified by the four corners included in the route request. For instance, in Figure 4.1, if node I receives the route request from another node, then it forwards the request to its neighbours, because it determines that it is within the rectangular request zone. However, when node J receives the route request, it discards the request, as node J is not within the request zone.

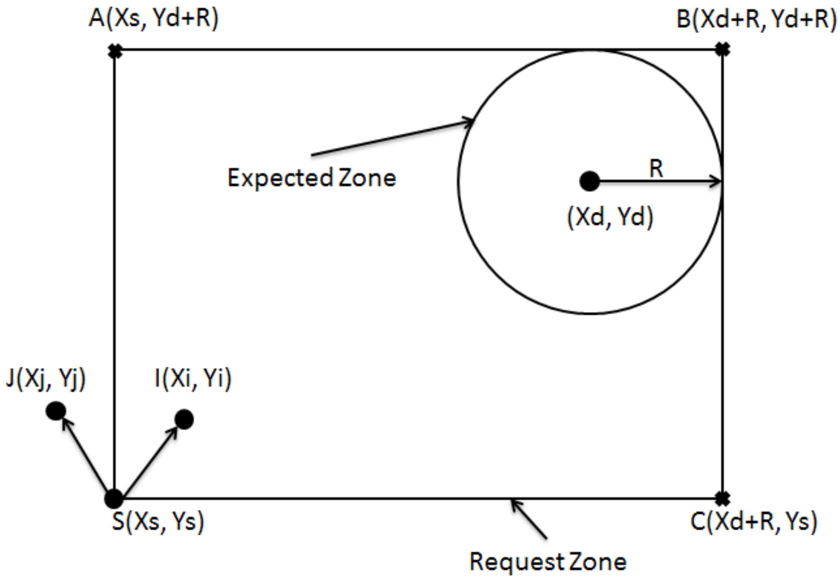

Figure 4.1: LAR route request zone
[37]

DSR [38] is a reactive routing protocol in which nodes request routing information only when needed. It is based on source routing concept, where the sender (S) constructs a source route in the packet's header to the destination (D). So S knows the complete hop-by-hop route to D and these routes are stored in a route cache. If S wants to send data packets to D, it will first search in its route cache a path to it and a route discovery process is initiated by S if such information is not available. During the route discovery process the network is flooded with route request (RREQ) packets. If a path is found, S will receive a route reply packet listing the route hops to D. DSR utilizes a route maintenance scheme which uses the data link layer acknowledgments to learn of any lost links. If any lost link was detected, a route error control packet is sent to the originating node. The node will remove that hop in error from the host's route cache, and all routes that contain this hop must be truncated at that point.

AODV [39] shares DSR's on-demand characteristics which means that it also discovers routes when needed via a similar route discovery process. Different from DSR, AODV uses traditional routing tables with only one entry per destination. To prevent routing loops and keep updated the routes, AODV uses sequence numbers maintained at each destination which are carried by all routing packets. In AODV the source nodes do not maintain a complete hop-by-hop path to reach the destination like in DSR. An important characteristic of AODV is the maintenance of timer-based states in each node, regarding utilization of individual routing table entries. A routing table entry is expired if not used recently. If no information of the destination is found in the routing table, the path discovery process starts by broadcasting a route request control message "RREQ" that propagates in the forward path. If a neighbour knows the route to the destination, it replies with a route reply control message "RREP" that propagates through the reverse path. Hello messages are used to maintain paths and detect if neighbours are still in connectivity range. A route maintenance scheme is used when a link is lost by initiating route request control messages.

The Dynamic MANET On-demand (DYMO) routing protocol [40], [41], [42] enables reactive, multihop unicast routing between participating routers. The DYMO protocol uses a route discovery and a route maintenance procedure. In the route discovery, RREQ packets are sent from S in the entire network through a hop-by-hop dissemination process. Each intermediate DYMO router will record a route to S. When D receives the RREQ, it responds with a RREP sent hop-by-hop toward S. Each intermediate DYMO router that receives the RREP creates a route to the target, and then the RREP is unicast hop-by-hop towards S. When S receives the RREP it means that routes are established between it and D. Route maintenance consists of two operations. The routes in used are preserved by extending their lifetimes when packets are successfully sent. To keep the routes fresh and react to changes in the network topology, links over which traffic is flowing are monitored. A route error (RERR) is sent toward S to indicate the current route to a particular D is invalid or missing when no path is found. S deletes the route when it receives the RERR packet and if it needs to send data to D again, it will need to perform a route discovery. Sequence numbers are used to ensure loop freedom and to avoid the use of stale routing information.

Bellman-Ford is a proactive algorithm [40], [41], [42] in which nodes maintain distance tables to all nodes from which the shortest path can be calculated between S and D. By periodically changing information with neighbouring nodes the information from the tables is always updated. Each node will maintain a table to all nodes from the network with different costs such as the distance, time to transmit on the path, number of hops, latency, number of outgoing packets, etc.

The Qualnet simulation topology and results are presented further for a mobile sink which follows the MMM and were the above described five routing protocols are used to send data from S to D.

4.3 Simulation topology and results

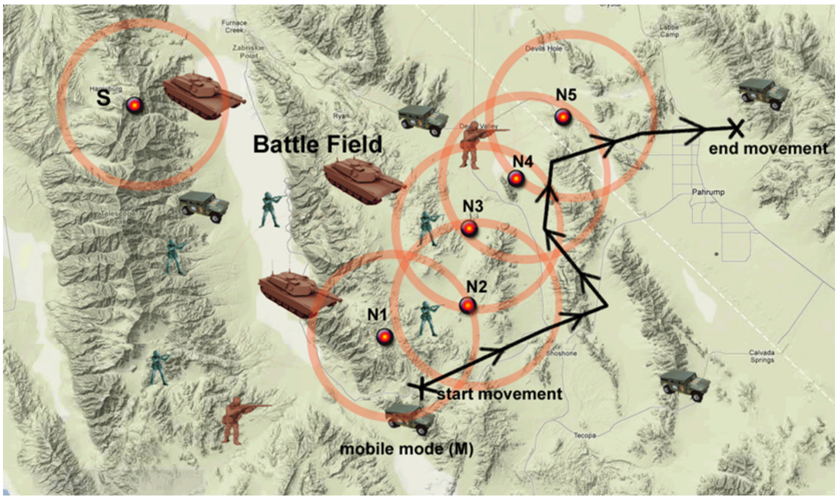

Figure 4.2: WSN military application scenario proposal

For the Qualnet simulations, a military application is considered (Figure 4.2) where, the mobile sink D travels at the WSN periphery to collect data. In [44] a similar scenario is discussed but a delay tolerant application is considered. While the study in [44] assumes a fixed delay tolerant level for each WSN node, in the currently proposed scenario the information needs to reach D very quickly, so no delays are permitted. Military applications have very stringent requirements when it comes to real time data delivery so it can be considered a DIN application. Using a DTMSN routing protocol is not a solution even if the WSN has DTMSN characteristics. A DTMSN routing protocol is designed based on the DTN characteristics received from the application itself. However, for our proposed scenario, the DTN characteristics are not known nor are they triggered by the application but by the WSN scenario.

A simple grid topology where 20 nodes are uniformly distributed like it is presented in Figure 4.3 is used. Each node has a transmission range of 430 meters. The distance between nodes was chosen to be 424 meters so beacon collisions can be

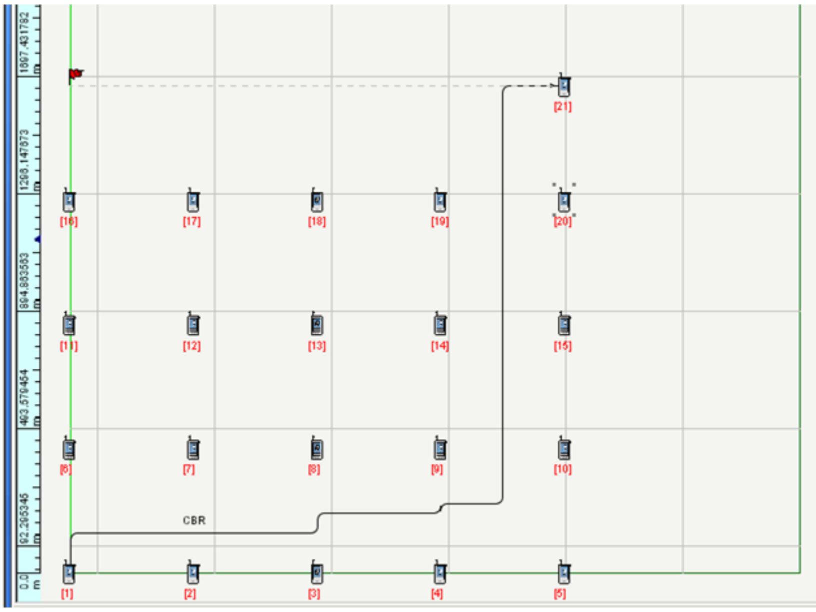

Figure 4.3: Qualnet simulation scenario

avoided. The network area has a rectangular shape where the data source (S) is the node from the bottom left-hand corner (node 1) and the mobile destination D (node 21) is the node from up right-hand corner. A constant bit rate (CBR) application layer is considered which generates constant traffic during the entire simulation. LAR, AODV, DSR, DYMO and Bellman-Ford are the routing protocols used to send data from S to D. The MAC 802.11 is used. Besides specifying the routing protocol, nodes position and MAC layer all the other parameters are the ones defined by default in Qualnet and these were not changed.

Four different scenarios are used as follows:

- In the first scenario S sends one data packet per second to a static D for a 55 seconds time period. D receives 54 data packets for each of the five routing protocols (the first packet is lost because of the 1 second set up time of the wireless sensor network). For this first scenario all the five routing protocols manage to send all the data to the mobile node. The path which is chosen by each protocol is presented in Figure 4.4. This proves that the Qualent

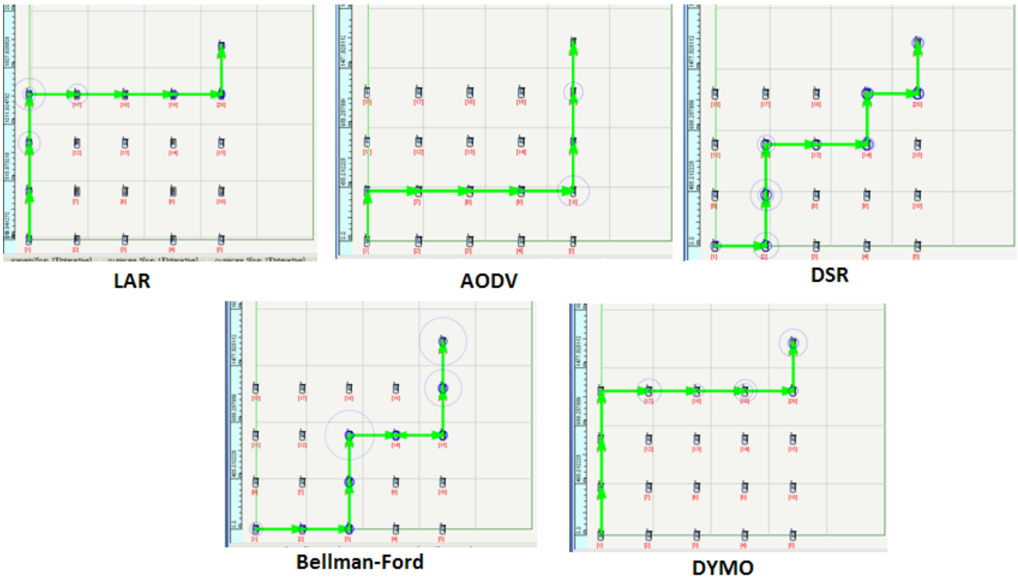

Figure 4.4: Selected paths by routing protocols for a static sink

simulation was done properly and that one second is sufficient time for the WSN setup.

- For the second scenario a mobile D is considered which moves with a speed of 30m/s following the Path 1 as it is presented in Figure 4.5. The distance between D and WSN is noted d(D,WSN) and is chosen to be 0 meters. For the entire simulation each routing protocol sends 54 packets, with 1 packet less when compared with the static scenario. This packet is lost because time is lost searching new routes when D exits the communication range of the periphery nodes.

- Because the nodes communication range (R) is 430 meters, the maximum distance that can be chosen between D and WSN is considered to be around 424 meters. It is not considered to be R because we need to have time for a path to be found and some packets sent before D exits its neighbour communication range. For the third scenario Path 2 is considered for the mobile D as presented in Figure 4.5 where d(D,WSN) = 244 meters was chosen as an intermediary distance between 0 meters and 424 meters. It can be seen that for this scenario *D* travels all the time connected with the WSN but the total time that travels in the WSN range is smaller when compared with Path 1 (*D* leaves sooner the coverage area of node 16). For this scenario LAR sends 51 data packets, AODV and DYMO 52, DSR 49 and Bellman-Ford 43.

- For the fourth scenario *D* moves following Path 3 (d(D,WSN) = 424 meters as it is presented in Figure 4.5). It can be seen that the for this case the total time that D travels disconnected with the WSN is maximum. During its movement, the mobile node will join first the network for a short period of time through node 20, it will then exit the coverage area of the network and move toward node 19 coverage. It will join node 19 for a short period of time and move through it. It then moves through node 18 and so on until it will reach node 16. In this scenario Bellman Ford and DSR fail to successfully send any data packets to the mobile node because the time that D is connected with the WSN throughout its movement is to small for a path to be found. AODV sends 3 out of 55 data packets, DYMO sends 8 and LAR 43. From these simulation results it can be observed that LAR sends 79% of data to the mobile node however the first packet that reaches the mobile node is at 43.03 seconds (of the 55 seconds simulation period).

In Figure 4.6 the results which are obtained by sending data packets to the mobile node as the distance between it and the network is varied, are presented. In Figure 4.7 the average end to end delay (AETED) (the average time it takes a data packet to reach the destination) for LAR is presented. It can be observed that this delay increases when the distance between the mobile node and the WSN coverage area increases. For the worst case scenario, when the mobile node is at a maximum

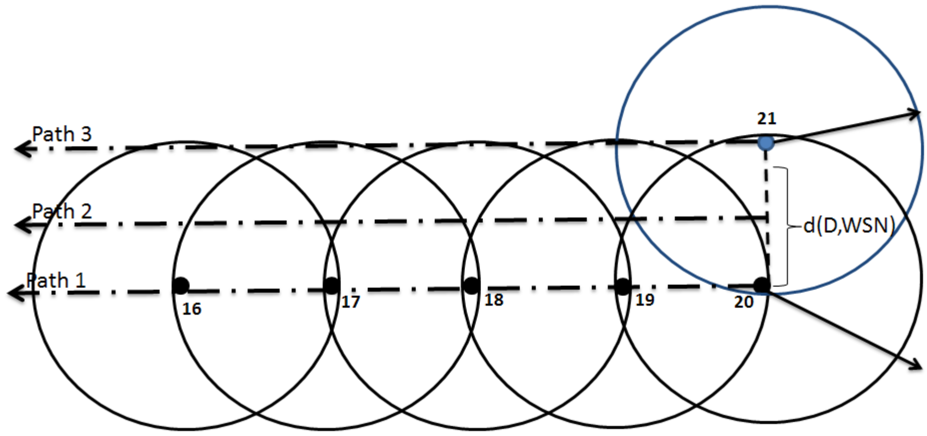

Figure 4.5: Different paths used for D

distance from the WSN area and the connectivity time is very small in comparison with the non connectivity time, the average end to end delay is around 21 seconds. This means that for time sensitive applications none of the five routing protocols considered can be used when the mobile sink follows the MMM. Because of this a new routing protocol needs to be developed which can support this model.

The speed of the mobile node influences the connectivity time with the WSN coverage area. Because of this, new simulations were carried out varying the speed of the mobile node between 5 m/s and 30 m/s. The distance between the mobile node and the WSN coverage area was set to 424 meters (the maximum distance). The results obtained are presented in Figure 4.8. The packet delivery ratio is kept under a satisfactory value if the speed of the mobile is varied up to 30m/s. LAR performs well and manages to send between 40 and 55 data packets (from a total of 56 data packets) during the speed variation of the mobile node. Figure 4.9 further presents the average end to end delay for LAR. From these simulations it can be concluded that none of the five routing protocols manage to send an acceptable number of data packets (over 90%) with an acceptable average end to end delay (i.e. less then 5 seconds) when the mobile node follows the marginal mobility model with a varying

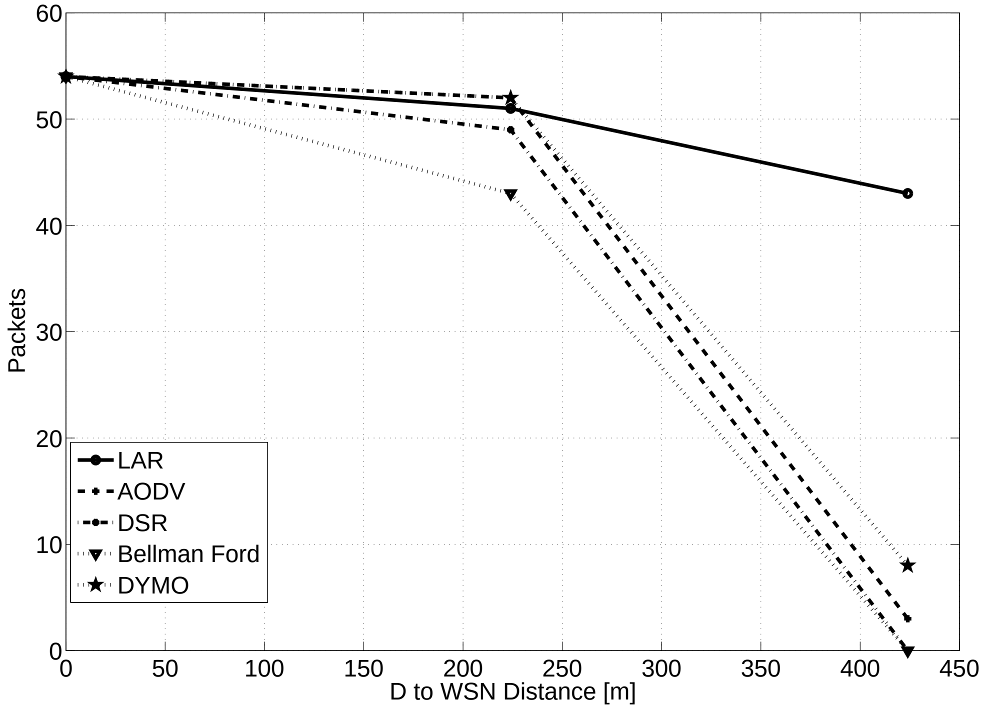

Figure 4.6: Packets received when the *D* to WSN distance is varied

speed and distance between it and the WSN coverage area. The MAC and routing layer interactions are discussed further.

4.3.1 MAC and routing layer interactions

In [125] it is shown that MAC level interactions play the primary role in determining the behavior of chains in a multi-hop wireless network. Two interesting observations are made:

1. MAC level interactions play a primary role in how multiple chain interactions. Other factors such as contention unfairness play a smaller role.

2. A chain that is well behaved with respect to self-interference is more immune to interference from another chain.

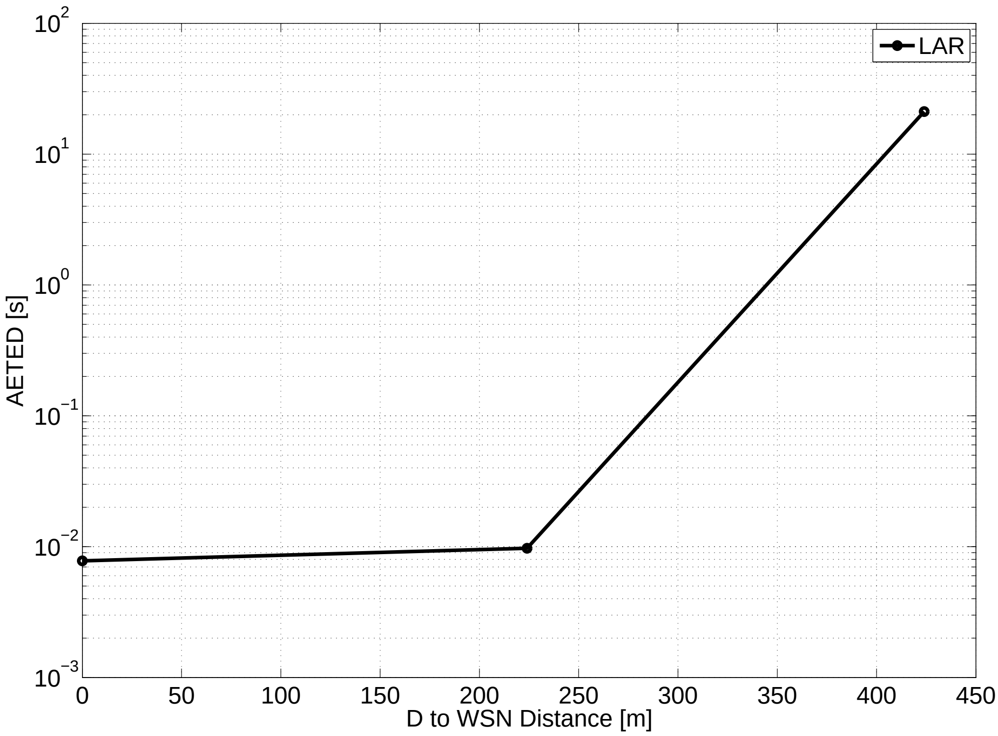

Figure 4.7: LAR AETED when the *D* to WSN distance is varied

The effect of mobility and interaction between various input parameters on the performance of protocols designed for wireless ad-hoc networks is studied in [51]. Authors' important objective is to analyze the interaction of the routing and MAC layer protocols under different mobility parameters. Three basic mobility models are used: grid mobility model, random way-point model and exponential correlated random model. AODV, DSR and LAR are the protocols used in the paper together with MACA, 802.111 and CSMA MAC protocols. Performance parameters like throughput, latency, number of packets received and long term fairness are analyzed. A comprehensive simulation based experimental is conducted to characterize the interaction between MAC, routing protocols, nodes' speed and injection rates in mobile ad-hoc networks. The evaluation criteria's used in this paper are: latency, total number of packets received throughput and long term fairness. Some main

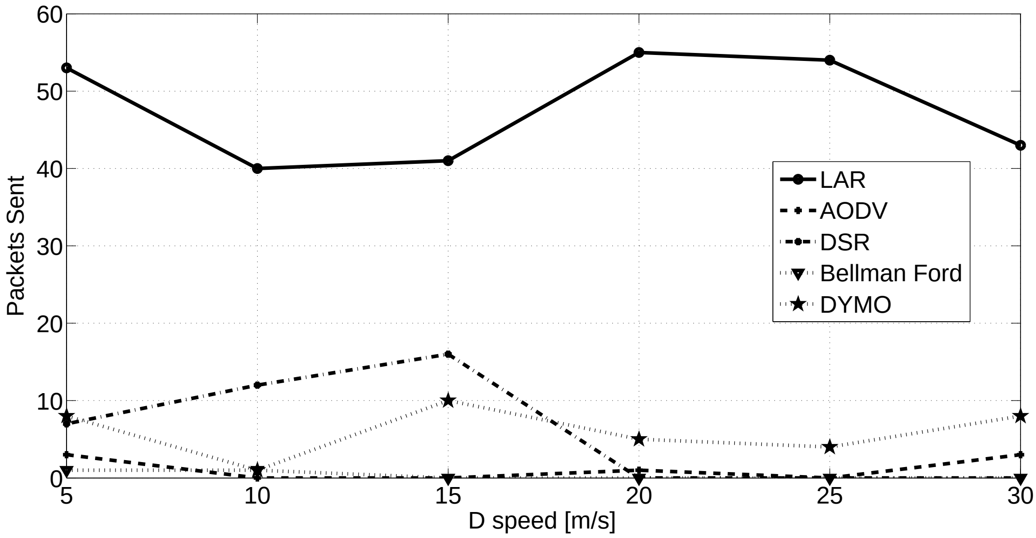

Figure 4.8: Packets sent by LAR, AODV, DSR, Bellman Ford and DYMO when D's speed is varied

conclusion of this paper can be summarized as follow:

- The performance of the network varies widely with varying mobility models, packet injection rates and speeds.

- MAC layer protocols interact with routing layer protocols so it is not meaningful to speak about a MAC or routing layer in isolation. These interactions are measured by the variation in the number of control packets generated by each layer. The results show that the routing protocol can significantly affect the MAC layer protocol and vice-versa. The pats that are chosen by the routing protocol induce a virtual network by exciting the MAC protocols at particular nodes. On the other hand, contention at the MAC layer can cause a routing protocol to respond by initiating new route queries an routing table updates. These interactions lead to trade-offs between the amount of control packets generated by each layer.

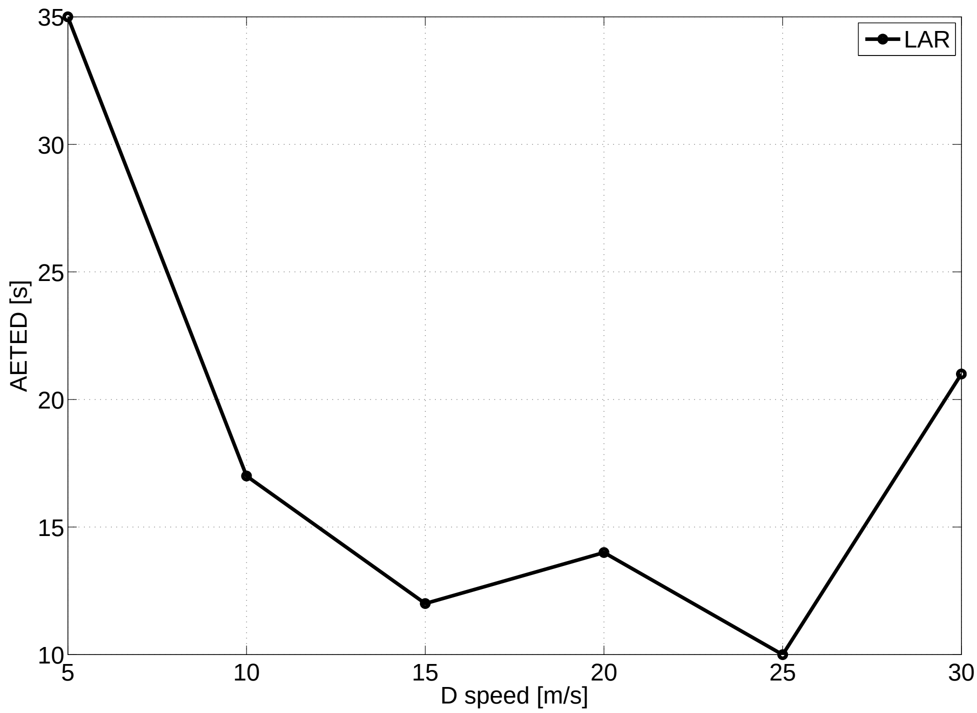

Figure 4.9: LAR AETED when *D*'s speed is varied

It is analysed further how different MAC protocols can influence the overall network performances when the same routing protocol is used. For the following simulations, the distance between *D* and WSN is set to a maximum value of 424 m and LAR is used for routing. The following MAC protocols are considered: MACA, CSMA, ALOHA, 802.11 and 802.11e. For TDMA no results were obtained. The total number of sent packets are presented in Figure 4.10 and the AETED in Figure 4.11

Based on this simulations it can be seen that the MAC protocol influence the total data packets sent to the mobile node. For the MMM the best results are obtained using the MACA protocol for MAC and LAR for routing. However the AETED is still not acceptable. It can be said that, based on the results obtained, the MAC protocol influence the routing protocol so in order to have the best results a perfect combination between these two is needed.

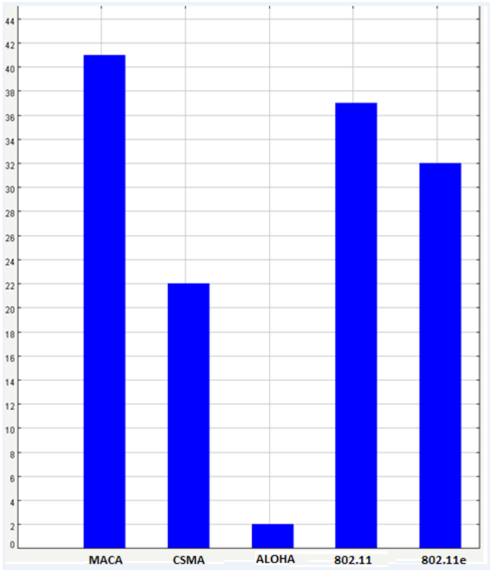

Figure 4.10: Packets sent by LAR when different MAC protocols are used

4.4 Conclusion

It was shown through Qualnet simulations that five MANET routing protocols cannot support a mobile sink which follows the MMM. For these simulations a simple WSN scenario was proposed where 20 nodes are uniformly distributed. The mobile node is considered to be placed in one corner of the WSN topology. For the simulations the distance between the mobile node and WSN coverage area was varied together with its speed. The mobile node receives data packets from the network, where the node placed in the opposite corner is considered to be the data source.

From the simulations carried out the following conclusions can be drawn. The distance between the mobile node and the WSN coverage area dramatically influences the total packets that are sent to D Figure 4.6. The connectivity time is influenced by the distance. When the distance is 0 meters the connectivity time is at a maximum so the routing protocols can deliver all the data packets to the mobile node. As the distance between the mobile destination node and the WSN coverage area

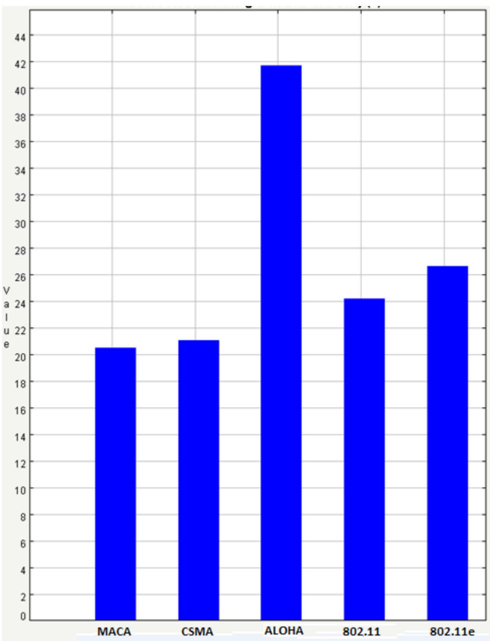

Figure 4.11: AETED obtained by LAR when different MAC protocols are used

increases, the time that the destination node stays connected with the rest of the network nodes decreases. In this case the packet delivery ratio drops significantly for all the routing protocols except LAR. However the large values obtained for the AETED for LAR in Figure 4.7 confirm that this routing protocol cannot satisfactorily support the MMM. The speed of the mobile node also influences the total number of sent packets as seen in Figure 4.8. All the five routing protocols fail to obtain a good packet delivery ratio or a small end to end delay when the speed of the mobile node is varied between 5m/s and 30m/s. It can be concluded based on the simulations carried out in this paper that AODV, DYMO, DSR, LAR or Bellman Ford do not adequately support the MMM so the assumptions made in [8] are now validated through simulations.

In the next chapter the Peripheral Routing Protocols (PRP) which is a new protocol proposed to support the MMM is described.

5 PRP description

In the previous chapter it was shown that different MANETs routing protocols have a poor performance for a WSN scenario in which the source (S) sends data packets to a mobile sink (D) which resides at the WSN periphery and which follows the MMM model. The Peripheral Routing Protocol (PRP) is a routing protocol proposal designed to obtain good results for this type of scenario. In this chapter details are provided about the necessity and functionality of this protocol such as: the conditions under which PRP is required for use (Section 5.1), the method of calculation for the next neighbour ID of D (Section 5.2), the calculation of the time duration that D travels outside the WSN (Section 5.3) and the recovery measures PRP uses to save power for worst case scenarios (Section 5.4).

PRP is a hybrid routing protocol: it uses a routing protocol recommended for MANETs (such as LAR) to search for a path and send data packets from S to the nodes at the WSN periphery and a peripheral routing algorithm (noted PRA) to estimate when and where the D will rejoin the WSN. So it can be stated that PRP is a combination of LAR and PRA. PRP needs to share both the mobile routing protocol characteristics recommended for MANETs (for the high connectivity part of the network it uses LAR) and DTMSNs routing protocols characteristics (for the low connectivity part it uses PRA). PRP improves on LAR by providing a LAR-enabled WSN with guidance for the zone request definition to help with path discovery and pre-emptive sending of packets to the estimated neighbour. A flow chart is presented in Figure 5.1 to explain how PRP works; it shows the blocks related to LAR in black colour and the blocks related to PRA in blue:

- **LAR:** S uses LAR to find the shortest path to D and send data packets. For this, route request packets (RREQs) are sent in an area which is calculated based on S and D coordinates [126].
- **PRA:** When D exits the WSN, its connectivity factor and probability of switching direction of movement are calculated. If both values are smaller than a certain threshold, then the next neighbour of D is estimated. S will find a path to the next neighbour of D using LAR and will start sending packets to its buffer. When D re-joins the WSN, its neighbour will send the packets held in the buffer. D will start to receive data packets directly from S again when the buffer is empty.

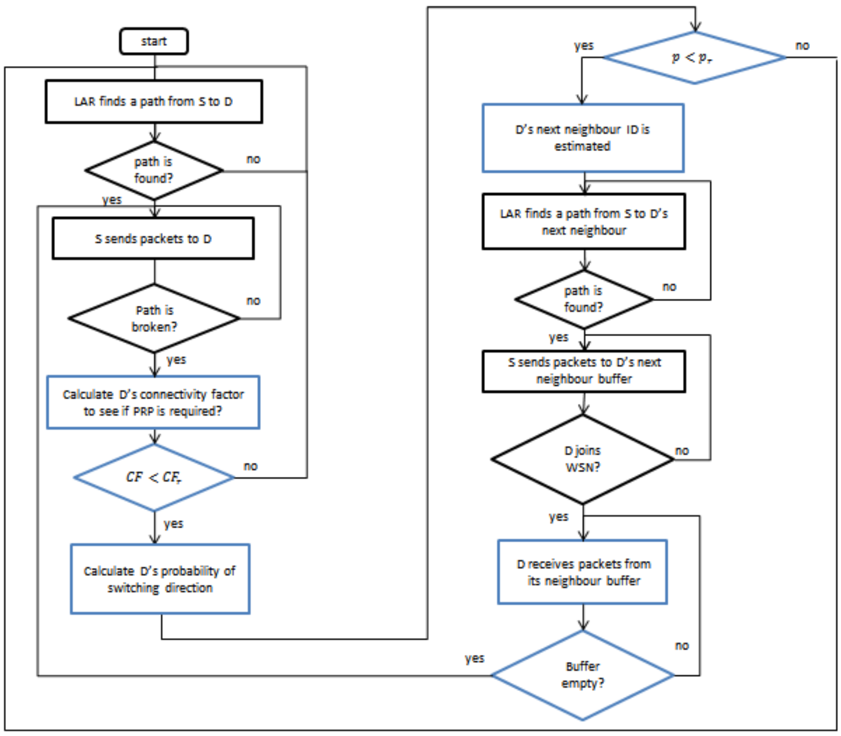

Figure 5.1: PRP flow chart

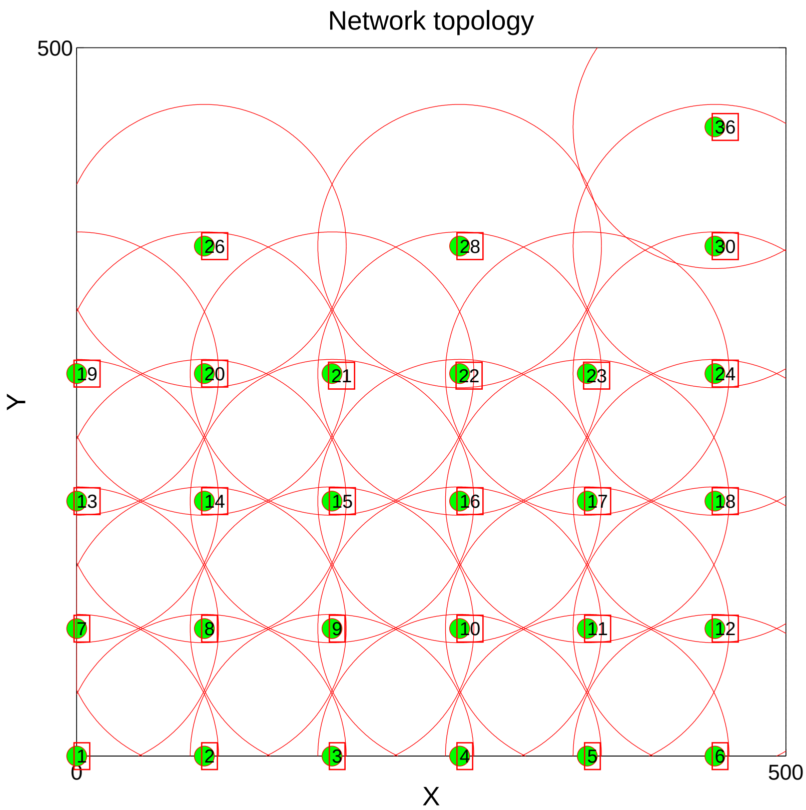

Figure 5.2: WSN Grid Topology

PRA uses information for D such as the connectivity factor, direction of movement and the probability of switching direction to estimate the next neighbour of D. The calculation of these values is explained further through the use of a simple grid topology which is presented in Figure 5.2 (it has a grid spacing of 90 m and 100 m node range). Node 1 which is the S needs to send data packets to node 36, which is the D. D moves from right to left in the northern part of the WSN, following the MMM.

When PRP is needed to be used is discussed in the next section.

5.1 When PRP is used

This section explains when a peripheral neighbour of D can decide that PRP needs to be used. Figure 5.3 aids in the presentation of the conditions a node needs to met in order to reside at the WSN periphery. When the WSN is set up, a localisation process begins and all nodes are informed of their positions (coordinates) in the WSN [127]. Any localisation process may result in location errors which means that each node position is an estimation. Each node will gather information about its own coordinates and that of the other nodes.

If it is considered that $N_{WSN} = \{N_i\left(x_i, y_i\right),\ i = 1, n\}$ represents all the nodes from the WSN area, then for each node N_a, $N_a \subset N_{WSN}$ which is not at the periphery means that it will need to have nodes in four geographic directions: north-east (NE), south-east (SE), south-west (SW) and north-west (NW). For this, N_a will search to see if $\exists\ N_b$, $\forall b = 1, n$, for which at least one condition from the Eq. 5.1 is not satisfied:

$$Cond = \begin{cases} x_b > x_a; \quad y_b > y_a;\ NE\ condition \\ x_b > x_a; \quad y_b < y_a;\ SE\ condition \\ x_b < x_a; \quad y_b < y_a;\ SW\ condition \\ x_b < x_a; \quad y_b > y_a;\ NW\ condition \end{cases}. \tag{5.1}$$

If at least one condition from Eq. 5.1 is not satisfied then it means that node N_a is a periphery node and it will be in the WSN geographic direction for which the condition was not satisfied. For example in Figure 5.3 a WSN network is considered. It can be seen that for N_2, the *NE condition* is not satisfied so it means that N_2 is a periphery node which resides at the north-east WSN periphery (NE). If two conditions from Eq. 5.1 are not satisfied, for example *NE condition* and *SE condition*

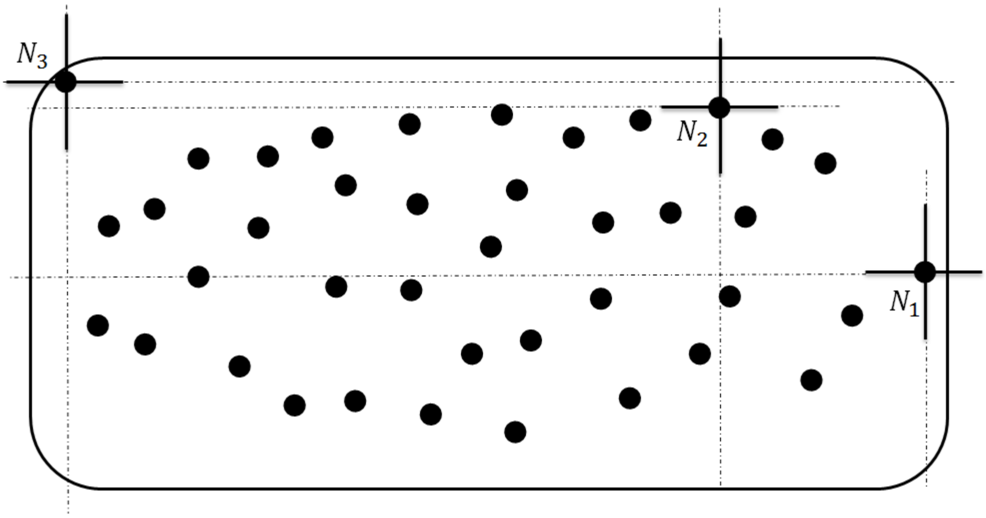

Figure 5.3: WSN periphery nodes

(N_1 in Figure 5.3) then it means that N_1 is a node which resides at the WSN East periphery (E). If three conditions are satisfied (for example node N_3), it means that the node is placed at the WSN periphery in a corner.

For irregular WSN scenarios (more realistic shapes, as depicted in Figure 5.4 a), if the conditions presented by Eq. 5.1 are tested for N_1, it will be concluded that N_1 is not a periphery node because it has nodes in all four geographic directions. For this, Eq. 5.1 can be rewritten as:

$$Cond = \begin{cases} x_a + \rho > x_b > x_a; \quad y_a + \rho > y_b > y_a; \ NE\ condition \\ x_a + \rho > x_b > x_a; \quad y_a + \rho < y_b < y_a; \ SE\ condition \\ x_a + \rho < x_b < x_a; \quad y_a + \rho < y_b < y_a; \ SW\ condition \\ x_a + \rho < x_b < x_a; \quad y_a + \rho > y_b > y_a; \ NW\ condition \end{cases}, \tag{5.2}$$

where ρ is considered to be the perimeter length in which the nodes are searched and it can be chosen based on the WSN size and node communication range. Figure 5.4 b) shows the area determined by $x_a + \rho$, $x_a - \rho$, $y_a + \rho$ and $y_b - \rho$ (noted H) in which N_1 will search to see if there are nodes in NE, NW, SE and SW. It can be observed that no nodes are detected in the SE region which makes N_1 a periphery node. This means that Eq. 5.2 is more suitable to determine the peripheral nodes for realistic WSN topologies.

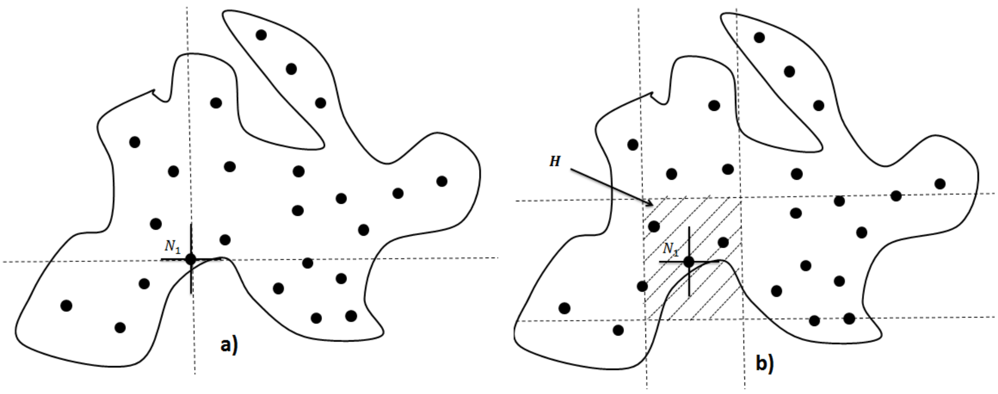

Figure 5.4: a) Periphery node scenario for Eq.5.1; b) Periphery node scenario for Eq. 5.2

Once the periphery nodes are determined, the next step will be to establish if PRP needs to be used. It is considered that D records and saves information regarding previous connections with the network during its movement: the amount of time it was able to communicate with the rest of the nodes (noted T_C – connectivity time) and the period of time it was out of the range of its neighbours (noted T_D – disconnectivity time). Based on this information, its neighbours will calculate the connectivity factor (C_F) using Eq. 5.3:

$$C_F = \frac{T_C}{T_C + T_D} * 100. \tag{5.3}$$

C_F is 100% when D communicates with the network without interruption during its movement. C_F will be compared with C_T which is a threshold value established by the user and it is influenced by the application requirements, network topology and size, etc. For example, an application can demand that D is connected with the network 95% of the time so that the dedicated data is quickly delivered. In this case $C_T = 95\%$ and if $C_F < C_T$ it means that the sink node does not keep a good connection with the network, so the PRP needs to be used by the peripheral neighbours of D to send the data packets. Otherwise, if $C_F > C_T$ a MANET routing protocol will be used such as LAR.

It is explained further how D's next neighbour ID is calculated.

5.2 D's next neighbour ID estimation

Once D's neighbour establishes that PRP needs to be used, based on the C_F value, D's direction of movement and time that D travels disconnected until it will re-join the WSN (T_D) needs to be calculated. Based on these values S will know which is D's next neighbour ID (through which node D will re-join the WSN). S will send packets to D's next neighbour buffer only for T_D seconds to avoid the situation in which D will not re-join the WSN through the estimated neighbour or it will not re-join the WSN at all. Although their location is known, D's current neighbour needs to determine the D's next neighbour node ID. Also, because of the location errors, T_D will be an estimated value.

D can have restrictions when it comes to power, hardware or communication time and because of this two scenarios are possible when D needs to send hello packets after it breaks the communication with its neighbour:

1. **No constraints for** D**.** In this case D will start to send hello packets to rejoin the WSN as soon as it breaks its communication with its neighbour.
2. D **has constraints**. D needs to start and send hello packets to rejoin the WSN when it reaches its next neighbour communication area. For this case T_D needs to be calculated.

For both scenarios, because D can change its direction of movement at any time and because the location error can influence the value obtained for T_D, these values are estimated values.

D's direction of movement needs first to be calculated to estimate D's next neighbour ID. It is assumed that a periphery node i (N_i) knows if its position is at the network periphery (as it was shown in Section 5.1). P represents the set of peripheral nodes of the network: $P = \{P_N \bigcup P_S \bigcup P_E \bigcup P_W\}$, where P_N, P_S, P_E and P_W are the set of peripheral nodes from the north, south, east and west side of the network. $P_N = [N_{30}, N_{28}, N_{26}]$ for the scenario presented in Figure 5.2. Let X_i be the connectivity indicator between a peripheral node i (N_i) and D, where $X_i = 1$ if N_i is connected with D and $X_i = 0$ if it is not connected. A peripheral connectivity vector can be defined as $PCV = [X_i]$, $i \epsilon P$, and $PCV = \{PCV_N \bigcup PCV_S \bigcup PCV_E \bigcup PCV_W\}$, where PCV_N, PCV_S, PCV_E and PCV_W are the north, south, east and west connectivity vectors. If PCV_N^b, PCV_S^b, PCV_E^b and PCV_W^b are the binary values for PCV_N, PCV_S, PCV_E and PCV_W, we can define PCV_N^d, PCV_S^d, PCV_E^d and PCV_W^d as the values for PCV_N^b, PCV_S^b, PCV_E^b and PCV_W^b in decimal. In the scenario presented in Figure 5.2, at time t_1, D is connected with node 30 so $PCV_{N,t_1} = [001]$, $PCV_{N,t_1}^b = 001$ and $PCV_{N,t_1}^d = 1$. If at t_2 D is connected with node 28 and $PCV_{N,t_2} = [010]$, $PCV_{N,t_2}^b = 010$ and $PCV_{N,t_2}^d = 2$ so it can be said that D moved from the right to the left because $PCV_{N,t_2}^d \geq PCV_{N,t_1}^d$.

A neighbour of D, at a certain moment in time t_i, can decide if D is moving left or right by comparing $PCV_{t_i}^d$ with $PCV_{t_{i-1}}^d$. Let D_m be the direction of movement indicator where:

$$D_m = \begin{cases} 0, & mobile\ node\ moves\ to\ left,\ if\ PCV_{t_i}^d > PCV_{t_{i-1}}^d \\ 1, & mobile\ node\ moves\ to\ right,\ if\ PCV_{t_i}^d < DPCV_{t_{i-1}}^d \end{cases} . \tag{5.4}$$

If D switches the direction of movement frequently, knowing D_m is not sufficient for D's neighbours to inform S where to send the data (to either the peripheral node's neighbour on their left or on their right). Because of this we use D's probability of switching direction of movement at t_β, noted $P_{\alpha_{t_\beta}}$:

$$P_{\alpha_{t_\beta}} = \frac{\alpha_{t_\beta} - 1}{sum(M_{x_{t_\beta}}) - 1}; \tag{5.5}$$

$$M_{x_{t_\beta}} = \sum_{i=1}^{\beta} PCV_{t_i}; \tag{5.6}$$

where $M_{x_{t_\beta}}$ represents the sum of the connectivity vectors generated at different periods of time t_i, $i = 1, \beta$; $\alpha_{t_\beta} = max(M_{x_{t_\beta}})$ represents the element with the highest value and $sum(M_{x_{t_\beta}})$ represents the sum of all vector elements.If $P_{\alpha_{t_\beta}} < P_{\alpha_r}$, where P_{α_r} is a threshold value imposed by the application requirements, it means that D moves following the direction of movement estimated by D_m. A short example of how the probability of switching the direction of movement based on the scenario presented in Figure 5.2 is explained further. If it is assumed that at t_1 D (node 36) is connected with node 30, then the north peripheral connectivity vector at t_1 is $PCV_{N,t_1} = [0, 0, 1]$ and $M_{x_{t_1}} = [001]$. If, at t_2, D moves from right to left and joins the WSN through node 28, then $PCV_{N,t_2} = [0, 1, 0]$ and $M_{x,t_2} = [001] + [010] = [011]$. At t_3, D joins the WSN area through node 26, then $PCV_{N,t_3} = [1, 0, 0]$, and $M_{x_{t_2}} = [001] + [010] + [100] = [111]$. At t_4, D moves from left to right and enters the connectivity area of node 28, so $PCV_{n,t_4} = [0, 1, 0]$, and $M_{x,t_4} = [111] + [010] = [121]$. So, during its movement from t_1 to t_4, D moved from the connectivity areas of node 30 to node 28, then 26 and back to node 28. It moved 3 times and changed the movement direction once. So based on Eq. 5.5,

$P\alpha_{t_4} = 100 * \frac{2-1}{4-1} = 33$. It can be concluded that D has a probability of switching direction of 33%. The calculated probability is compared with a threshold value and based on the result, D's neighbour can estimate if D will re-join the WSN through its right or left peripheral neighbour.

Once D's direction of movement is known, D's neighbour can estimate through which node D will rejoin the WSN: through its peripheral neighbour from the left or from the right. It was previously explained that when D leaves the WSN area, its neighbour will inform S to send the data packets to the next neighbour of D. S will start sending packets to it until D will rejoin the WSN and it will start to receive packets from its neighbour buffer. In order to rejoin the WSN, D sends hello packets during the time it is disconnected. Sending hello packets will result in an additional energy consumption. If D does not have any energy constrains, then the hello packets can be sent as soon as D leaves the WSN until it will find its next neighbour. However, if D has energy constrains, sending hello packets only when it approaches its next neighbour communication range is crucial for saving power. For this, D will receive from its neighbour the time after it will need to send hello packets noted T_D.

In the next section it is explained how T_D is calculated.

5.3 How $\mathrm{T_D}$ is calculated

Only a two dimensional scenario is considered and it is explained how T_D is calculated with the help of Figure 5.5 where: D moves following the line l , N is the current D neighbour and M is the next D neighbour, $A(x_A, y_A)$ and $B(x_B, y_B)$ are the entry and exit points of D from N coverage area, $C(x_C, y_C)$ and $E(x_E, y_E)$ are the entry and exit points of D from M coverage area. When D breaks the communication with N (at point B), to calculate the time that D travels disconnected with the WSN (T_D), Eq. 5.7 is used:

$$T_D = \frac{d}{v}; \tag{5.7}$$

where d is the distance that D travels disconnected (between point B and C as it is shown in figure Figure 5.5) and v is the speed of D.

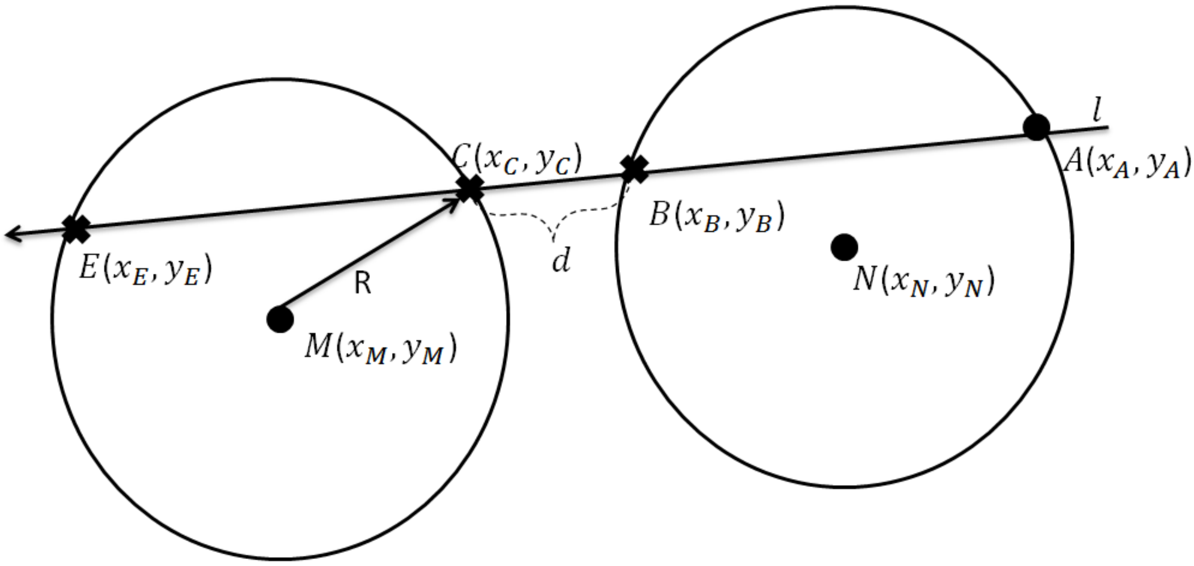

Figure 5.5: D's disconnected time between two peripheral nodes

To calculate d ($d = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2}$), D's neighbour (N) needs to estimate the x and y coordinates of B and C. It is assumed that D is equipped with a GPS so the coordinates of A and B are known. The coordinates of C are estimated using Eq. 5.8 and Eq. 5.9 :

$$x_C = \frac{-B + \sqrt{\Delta}}{2 * A}; \tag{5.8}$$

$$y_C = m * (x_C - x_B) + y_B; \tag{5.9}$$

where: $A = 1+m^2$; $B = -2*x_M - 2*P$; $C = x_M^2 + P^2 + R^2$; $P = m*x_B - d + y_M$; $\triangle = B - 4*A*C$. All the following parameters are known by D's neighbour: x_M and y_M are the coordinates of D's next neighbour (M); x_B and y_B are B coordinates; m is the slope of the straight line l; R is the range of the circle with the center in M. m is calculated using Eq. 5.10:

$$m = \frac{y_A - y_B}{x_A - x_B}; \tag{5.10}$$

where x_A and y_A are A coordinates, x_B and y_B are the coordinates of B.

The mathematical formula for x_C and y_C is obtained by solving a system of two equations as follows:

- For obtaining y_C: if m is l slope, and $X(x_0, y_0) \in l$, then the equation of l is: $y - y_0 = m * (x - x_0)$. In Figure 5.5 $B(x_B, y_B) \in l$ so the equation becomes: $y - y_B = m * (x - x_B)$. Also, the point $C(x_C, y_C) \in l$ so its coordinates need to respect l equation, this means that $y_C - y_B = m * (x_C - x_B)$ which result in Eq. 5.9 for y_C

- For obtaining x_C: the general equation of a circle $C(O, R)$, where $O(x_0, y_0)$ is the center and R is the range, is: $(x - x_0)^2 + (y - y_0)^2 = R^2$. For the node $N(x_N, y_N)$ and communication range R, the circle equation becomes: $(x - x_N)^2 + (y - y_N)^2 = R^2$. Also, the point $C(x_C, y_C) \in C(N, R)$ so the circle equation becomes: $(x_C - x_N)^2 + (y_C - y_N)^2 = R^2$, where x_C is calculated (Eq.

5.8) by substituting y_C and for which we consider the value with $+\sqrt{\Delta}$. The other value with $-\sqrt{\triangle}$ is the x_E coordinate of point E and its y_E coordinate is calculated also with Eq. 5.9. E is D's exit point from M range.

After T_D and $C(x_C, y_C)$ are estimated, D's neighbour will be able to estimate where and when it will join the WSN. When D will leave the WSN area, its neighbour will inform S to send data packets to M. At this point D will have two options:

- to either send hello packets as soon as the communication between it and WSN breaks.
- or to send hello packets only when it will approach its next neighbour communication area.

Once D's neighbour decides that PRP needs to be used by comparing C_F with C_T (Section 5.1), D's next neighbour direction of movement and probability of switching direction are calculated (Section 5.2). If $P_{\alpha_{t_\beta}} < P_{\alpha_r}$ it means that D's direction of movement can be estimated and that D's neighbour can estimate where and when D will rejoin the WSN after it will leave its communication range.

PRP recovery measures are discussed in the next section for the scenarios in which $P_{\alpha_{t_\beta}} > P_{\alpha_r}$ or D will not join the WSN through its estimated neighbour at the estimated time.

5.4 PRP recovery measures

Recovery measures need to be considered in order to save power for the case when D does not re-join the WSN through its estimated neighbour or it does so after the estimated time. If $P_{\alpha_{t_\beta}} > P_{\alpha_r}$ it means that D's probability of switching direction of movement is high so D's neighbour cannot predict through which node D will re-join the WSN. In this case:

- D starts to send hello packets as soon as it leaves its neighbour communication range.
- When a hello packet is received by a peripheral node, it means that D re-joined the WSN. In this case, a route discovery process is started from the peripheral node to find a route between it and S. RREQ zones are used based on its coordinates and S coordinates similar to LAR.
- Once a route between S and D is found, S will start to send data packets to D.

If $P_{\alpha_{t_\beta}} < P_{\alpha_r}$, S is informed of which peripheral node to send data packets. But D can stop moving after it exits the WSN area or it can change the direction of movement and re-join the WSN through a different peripheral node. In order to avoid sending a large number of data packets to a node through which D will not re-join the WSN, the following procedure is considered:

- S also receives the T_D value (time that D will travel disconnected until it will rejoin the WSN) when it is informed of which peripheral node to send data until D will rejoin the WSN.
- Once D rejoins the WSN, a route reply message is sent to S by D's neighbour to confirm that D re-joined the WSN.
- If a route reply message is received by S after T_D seconds it means that D rejoined the WSN at the estimated location after the estimated time. In this case S will continue to send data packets to D's neighbour. If not, S will stop sending packets and it will wait to receive a route reply message (confirming that D re-joined the network through its predicted next neighbour) or a route request packet (which means that D re-joined the WSN through a different node.)

To summarize, this chapter has explained the following: that PRP is a combination of LAR and PRA; that PRP is considered to be used by D's neighbour; that D's next neighbour ID is estimated based on D's direction of movement so that S can send packets to it until D will rejoin the WSN; that D will start to send hello packets to save power and what safety measures are considered to avoid sending a large amount of data packets to a node through which D will not rejoin the WSN. The next chapter analyses and compares the equations which describe the throughput and energy spent by PRP and LAR.

6 PRP and LAR analysis

The equations which describe the throughput (T) and energy spent per transmitted packet (E) for LAR and PRP are analysed and discussed in detail in this chapter. In section 6.1 LAR and PRP equations which describe the total number of sent packets by S to D are presented and compared. The equations which describe the E for both LAR and PRP are presented in section 6.2. The impact of location errors on LAR and PRP performance is analysed in section 6.3 and the chapter ends with section 6.4 where conclusions are stated.

T is used to measure the amount of data that reaches D in a certain amount of time. It can be expressed as packets per second (as it will be used in this chapter) or kilobytes per second and is calculated with Eq. 6.1

$$throughput = \frac{Pckts}{T_{sim}}; \tag{6.1}$$

where $Pckts$ is the total number of packets received by D and T_{sim} represents the simulation time. The energy spent per transmitted packet will be noted E and it will reflect the energy spent on routing plus the energy spent to find a path from S to D.

The scenario considered is the one presented in Figure 5.2 where D moves in the north part of the WSN following the MMM. PRP and LAR mathematical equations which describe $Pckts$ and E are presented next with the help of Figure 6.1 for LAR and Figure 6.2 for PRP. D exits the communication range of node N through point B and will rejoin the WSN through node M at point C. The parameters that influence the $Pckts$ and E for both LAR and PRP are consequently determined.

6.1 Total number of packets received by D

6.1.1 LAR analysis

When LAR is used, S sends data packets to D only when D is connected with the network and a path from S to D has been found. When D breaks the communication with its neighbour, S is informed and starts a path discovery process to D by sending RREQ packets in the WSN area determined by both S and D coordinates. After a path has been discovered, S will start sending packets each t_s seconds. The first packet will reach D after $t_p * n_p$ seconds (t_p is the time that it takes for a packet to be sent between two hops from the path, n_p is the hop count of the path, $t_p = 1/p_r$ where p_r is the packet rate and $p_r = d_r/p_s$ where d_r is the data rate and p_s is the packet size). The number of packets that reach the destination is considered to be $Pckts_{LAR}$ and is calculated using Eq. 6.2 (if it is assumed that at least one packet will reach D, which means that $T_{IC} > \tau_L + T_{RREP} + t_p * n_p$):

$$Pckts_{LAR} = 1 + \left[\frac{T_{RC'}}{t_s}\right] = 1 + \left[\frac{T_{IC} - (\tau_L + T_{RREP} + t_p * n_p)}{t_s}\right]; \qquad (6.2)$$

where 1 represents the first packet that reaches D after $t_p * n_p$ seconds, t_s is the time that S waits between two data packets transmission, $T_{RC'} = T_{RC} - t_p * n_p =$

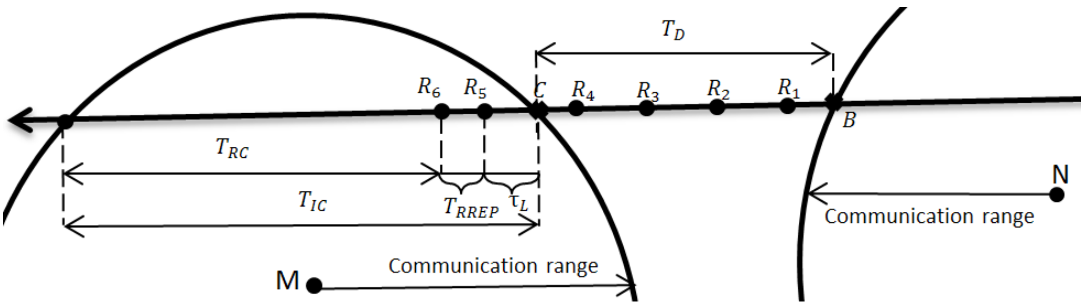

Figure 6.1: WSN scenario for LAR

$T_{IC} - \tau_L - T_{RREP} - t_p * n_p$ is the real connectivity time between D and its neighbour in which D receives packets from S at each t_s seconds. T_{IC} is the ideal (maximum) connectivity time between D and M, τ_L is time that D travels connected with its neighbour before it receives a RREQ packet from S and T_{RREP} is the time spent by D to send a RREP packet to S.

With the help of figure Figure 6.1 it is explained further how τ_L is calculated. It is assumed that D is leaving N's communication range and is heading to M's communication area. During this time, S sends RREQ packets every ξ_L seconds (ξ_L is the RREQ broadcast time window), where $\xi_L = T_{RREQ} + T_{wait}$, T_{RREQ} is the time that it takes for a RREQ packet to be sent by S to the furthest node from the RREQ zone, T_{wait} is the waiting time between two RREQ packets which is considered to save power and not flood the WSN with these packets. R_i, $i = 1..5$ are marked positions where D is when a RREQ is sent at each ξ_L seconds. R_5 is the position where D is found to be in M's communication range and R_6 when D starts to receive data packets from S after a route reply package is sent back to S. If T_D is the time that D travels disconnected between node N and M (between point B and C) then τ_L can be calculated with Eq. 6.3 :

$$\tau_L = \left(\left\lceil \frac{T_D}{\xi_L} \right\rceil + 1 \right) * \xi_L - T_D. \tag{6.3}$$

For the scenario in which D is all the time connected with the WSN ($T_D = 0$), only one RREQ packet will be sent to find a path from S to D (which means that $T_{wait} = 0$) so $Pckts_{LAR}$ becomes (calculated in Appendix 1):

$$Pckts_{LAR} = 1 + \left\lceil \frac{T_{IC} - (T_{RREQ} + T_{RREP} + t_p * n_p)}{t_s} \right\rceil . \tag{6.4}$$

It can be seen that LAR will not send any packets ($Pckts_{LAR} = 0$) when $T_{IC} < \tau_L + T_{RREP} + t_p * n_p$ and for the scenario in which $T_D = T_{wait} = 0$ when $T_{IC} \leqq T_{RREQ} + T_{RREP} + t_P * n_p$.

To determine the conditions for which LAR will send the maximum number of packets (which factors influence $Pckts_{LAR}$ performances), $Pckts_{LAR}$ needs to be maximized. It can be seen that $max\,(Pckts_{LAR})$ implies $max\,(T_{IC})$ and $min(\tau_L + T_{RREP} + t_p * n_p)$. However, the only parameter that can be changed by the user is T_{wait} so, $min(\tau_L + T_{RREP} + t_p * n_p) \rightarrow min(\tau_L) \rightarrow min(T_{wait})$.

$T_{wait} = 0$ is needed to maximize the number of data packets that LAR can send for the scenario depicted in Figure 6.1. This means that the waiting time between two RREQ packets needs to be 0. In this case, when D will break the communication with the WSN, RREQ packets will flood the network and by doing this, $Pckts_{LAR}$ is maximised but the energy spent on sending RREQ packets drastically increases. Hence, a compromise needs to be found between the desired $Pckts_{LAR}$ and E_{LAR} when T_{wait} is chosen. A PRP analysis is made further to determine the parameters which influence the packets sent by S to D.

6.1.2 PRP analysis

When PRP is used to send data packets from S to D, when D breaks the communication with the WSN, its former neighbour estimates D's next neighbour coordinates. It will then instruct S to find a path to D's next neighbour and S will start sending packets to it until D will rejoin the WSN. So, for PRP, the scenario depicted in Figure 6.1 becomes the one presented in Figure 6.2 where: D breaks N's communication range at point B and it is moving to M's communication area; with R_i, $i = 1..5$ are the marked positions where D is when it sends hello packets to rejoin the WSN at each T_{hello} seconds; at R_5 D will send a RREQ packet and it will receive a RREP packet from M. At this point, M will inform S to restart the data transmission after it will send all the buffered packets to D. At R_6, D starts to receive packets from M's buffer. The time spent between R_5 and R_6 is noted T_{ec} (time to established the connection between M and D) and it can be calculated as $T_{ec} = T_{rreq} + T_{rrep} + T_{inf}$ were: T_{rreq} is the time spent by D to send a RREQ packet when it is in M coverage area, T_{rrep} is the time spent by M to reply to D and T_{inf} is the time spent by M to inform the S to restart the data transmission when M's buffer will be empty. From R_7 to point E, D will receive packets from S again for T_{RS} seconds, after all the packets from M's buffer are sent.

Let $Pckts_{PRP}$ be the total number of packets sent by S to D. $Pckts_{PRP} = Pckts_b + Pckts_c$ where $Pckts_b$ is the total number of packets sent from M's buffer during T_{RB} and $Pckts_c$ is the total number of packets which are sent during T_{RS}.

$Pckts_b$ can be written using Eq. 6.5

$$Pckts_b = 1 + \left[\frac{T_{Dec'}}{t_s}\right]; \tag{6.5}$$

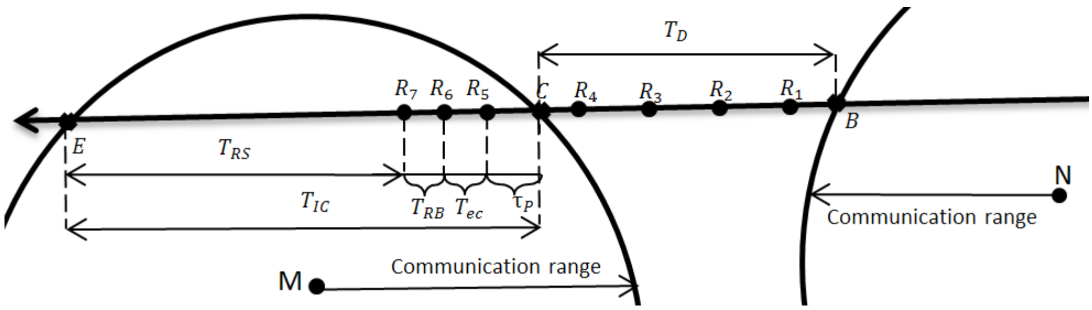

Figure 6.2: WSN scenario for PRP

where $T_{Dec'}$ is the time in which all the data packets reach M's buffer after t_s seconds, so $T_{Dec'} = T_{Dec} - t_p * (n_p - 1)$ where $t_p * (n_p - 1)$ is the time spent by the first packet to reach M's buffer. T_{Dec} is D's connectivity time, $T_{Dec} = T_D + \tau_P$ where T_D is the time that D travels between points B and C and τ_P is the time needed for M to detect D. $Pckts_c$ can be written using Eq. 6.6:

$$Pckts_c = \left\lfloor \frac{T_{RS}}{t_s} \right\rfloor; \tag{6.6}$$

where T_{RS} is the time in which D receives packets from S each t_s seconds so $T_{RS} = T_{IC} - \tau_P - T_{ec} - T_{RB}$. The time spent to empty M's buffer is $T_{RB} = Pckts_b * t_p$. τ_P is calculated with Eq. 6.7:

$$\tau_P = \left(\left\lfloor \frac{T_D}{\xi_P} \right\rfloor + 1 \right) * \xi_P - T_D; \tag{6.7}$$

where $\xi_P = T_{hello} + T_{wait}$ (which is the hello broadcast time window) .

$Pckts_{PRP}$ becomes:

$$Pckts_{PRP} = 1 + \left\lceil \frac{T_D + \tau_P - t_p * (n_p - 1)}{t_s} \right\rceil + \left\lceil \frac{T_{IC} - (\tau_P + T_{ec} + T_{RB})}{t_s} \right\rceil. \quad (6.8)$$

The conditions which need to be satisfied to validate $Pckts_{PRP}$ formula are: 1) $T_D + \tau_P \geq t_p * (n_p - 1)$. This means that D needs to be disconnected from the WSN at least $t_p * (n_p - 1)$ seconds which represents the time needed for one packet to reach M's buffer. Otherwise LAR will be used to send data packets from S to D. 2) $T_{IC} \geq \tau_P + T_{ec} + T_{RB}$ which implies that D is connected with M for sufficient time to receive all its buffered packets. Otherwise only a part from M's buffer packets are sent and $Pckts_{PRP} = \lceil (T_{IC} - \tau_p - T_{ec}) / t_p \rceil$.

To maximize $Pckts_{PRP}$ means to minimize τ_P which means to minimize T_{wait}. For $T_{wait} = 0$, D will start to send hello packets as soon as it leaves N's communication range. No waiting time will be used between hello packets which will lead to more energy consumption for D. This means that a compromise between $Pckts_{PRP}$ and E_{PRP} needs to be considered when choosing the value for T_{wait}. A LAR and PRP comparison is made next to determine which protocol sends more packets and under which conditions.

6.1.3 LAR and PRP comparison

In order to compare $Pckts_{LAR}$ with $Pckts_{PRP}$ the following equation needs to be solved: $Pckts_{PRP} - Pckts_{LAR} = 0$. If it is assumed that $\lceil T_{IC} - (\tau_L + T_{RREP} + t_p * n_p) / t_s \rceil = k_1 * t_s$, $\lceil T_D + \tau_P - t_p * (n_p - 1) / t_s \rceil = k_2 * t_s$ and $\lceil T_{IC} - (\tau_P + T_{ec} + T_{RB}) / t_s \rceil = k_3 * t_s$ then $Pckts_{PRP} -$

$Pckts_{LAR} = 0$ becomes (see Appendix 3 for details on the calculus):

$$T_D * \left(1 - \frac{t_p}{t_s}\right) = T_{ec} + \frac{t_p * [\tau_P - t_p * (n_p - 1)]}{t_s} - t_p - \tau_L - T_{RREP}. \tag{6.9}$$

For $^{t_p}/_{t_s} = {^1}/_x$, $Pckts_{PRP} - Pckts_{LAR}$ becomes (Appendix 3):

$$Pckts_{PRP} - Pckts_{LAR} = T_D * \left(1 - \frac{1}{x}\right) + \frac{t_p}{x} * (n_p - 2) + t_p * (2 * n_p - 1)\,.$$

It is known that S and D are more than one hop away ($n_p \geq 2$) and that D travels disconnected from the WSN ($T_D > 0$) which results in $Pckts_{PRP} - Pckts_{LAR} > 0 \iff Pckts_{PRP} > Pckts_{LAR}$. This means that, for the scenario in which D travels disconnected from the WSN and more then two hops are between S and D, PRP will send more data packets when compared with LAR.

Both LAR and PRP equations which describe the E are presented and compared in the next section.

6.2 Energy analysis for LAR and PRP

6.2.1 LAR analysis

The total energy spent by LAR is written with 6.10:

$$E_{LAR} = E_{LAR_{Routing}} + E_{LAR_{RREQ}}; \tag{6.10}$$

where $E_{LAR_{Routing}}$ is the total energy spent on routing and $E_{LAR_{RREQ}}$ is the total energy spent on RREQs.

However, $Pckts_{LAR}$ is the total number of packets sent by S to D, so $E_{LAR_{Routing}} = Pckts_{LAR} * e_{routing} * n_P$ where $e_{routing}$ is the energy spent to send a data packet between two nodes which are one hop away and n_p is the path length between S and D. The number of route requests sent by S in T_D seconds can be calculated with Eq. 6.11.

$$n_{rreq} = \left\lfloor \frac{T_D}{\xi_L} \right\rfloor + 1. \tag{6.11}$$

$E_{LAR_{RREQ}}$ it is written using Eq. 6.12:

$$E_{LAR_{RREQ}} = e_{rreq} * n_n * \left(\left\lfloor \frac{T_D}{T_{RREQ} + T_{wait}} \right\rfloor + 1 \right); \tag{6.12}$$

where n_n is the total number of nodes from the RREQ area in which the RREQs packets are sent and e_{rreq} is the energy spent on one RREQ.

In can be seen that in order to minimize E_{LAR} we can minimize $E_{LAR_{RREQ}}$ so we need to maximize T_{wait}. However $Pckts_{LAR}$ is also reduced if T_{wait} is maximized (as it was discussed in Section 7.1.1) so the value for T_{wait} needs to be chosen taking into account both $Pckts_{LAR}$ and E_{LAR} values that the application needs to satisfy. E equations for PRP are presented next.

6.2.2 PRP analysis

The total energy spent by PRP can be written using Eq. 6.13

$$E_{PRP} = E_{PRP_{Routing}} + E_{PRP_{hello}} + E_{S_{RREQ}}; \tag{6.13}$$

where $E_{PRP_{Routing}} = Pckts_{PRP} * e_{routing} * n_P$ is the total energy spent on routing, $E_{PRP_{hello}} = n_{hello} * e_{hello}$ is the energy spent by D on hello packets and $E_{S_{RREQ}}$ is the energy spent by S to find a path to D next neighbour. n_{hello} is the number of hello packets sent by D when it travels disconnected with the WSN and e_{hello} is the energy spent on sending one hello packet. n_{hello} is calculated with Eq. 6.14:

$$n_{hello} = \left\lfloor \frac{T_D}{\xi_P} \right\rfloor + 1; \tag{6.14}$$

where $\xi_p = T_{hello} + T_{wait}$. $E_{S_{RREQ}}$ is expressed by Eq. 6.15:

$$E_{S_{RREQ}} = n_m * e_{rreq}; \tag{6.15}$$

where n_m is the number of nodes which are in the RREQ area defined by the coordinates of S and D next neighbour.

$E_{PRP_{hello}}$ can be written as:

$$E_{PRP_{hello}} = \left(\left\lfloor \frac{T_D}{T_{hello} + T_{wait}} \right\rfloor + 1 \right) * e_{hello}. \tag{6.16}$$

It can be seen that in order to minimize E_{PRP} we can minimize $E_{PRP_{hello}}$ so we need to maximize T_{wait}. However, this will also lead to reducing the connectivity time between D and its neighbour. This will lead to a $Pckts_{PRP}$ decrease, as it was discussed in Section 7.1.2. So, a compromise between $Pckts_{PRP}$ and E_{PRP} needs to be taken into account when choosing T_{wait}. A comparison between LAR and PRP is made next to determine which protocol is more energy efficient.

6.2.3 LAR and PRP comparison

Both LAR and PRP use the shortest path between S and D. It will be analysed which protocol uses more energy if the same number of data packets are sent ($Pckts_{LAR} = Pckts_{PRP}$ which means that $E_{PRP_{Routing}} = E_{LAR_{Routing}}$). For this analysis the following scenario is considered: $T_D \neq 0$, $T_{wait} = 0$, and T_D is a multiple of $n_p * t_p$ meaning that $T_D = m * n_p * t_p$, $n_m = n_p + n_r$ and $n_m + n_o = n_n$. Following the calculus presented in Appendix 4 it can be seen that:

$$E_{LAR_{RREQ}} = E_{PRP_{hello}} + E_{S_{RREQ}} + e * (m * n_r + m * n_o + n_o - 1) . \qquad (6.17)$$

This means that for the scenario when D travels disconnected from the WSN, the energy spent on RREQs by LAR is bigger than the energy spent by PRP on hello packets plus the energy to find a path from S to the next neighbour of D.

Figure 6.3 is used for a better understanding of this conclusion where:

- S sends data packets to D which follows the MMM model.
- D breaks the communication with the WSN and is heading to M's communication range.

- n_n represents the total number of nodes in the RREQ area defined by S and D's coordinates in which S will send RREQ packets when LAR is used and D exits the WSN area.

- n_m represents the total number of nodes in the RREQ area in which S sends RREQ packets to find a path to M when PRP is used and D leaves the WSN area. It can be seen that $n_m + n_o = n_n$.

- n_p is the number of nodes which form the path between S and D. It can be seen that $n_m = n_p + n_r$.

From the above analysis it can be concluded that, for a scenario in which D travels disconnected from the WSN, PRP uses less energy to send the same number of data packets when compared with LAR. In the next section it is analysed how both LAR and PRP performance is influenced by location errors.

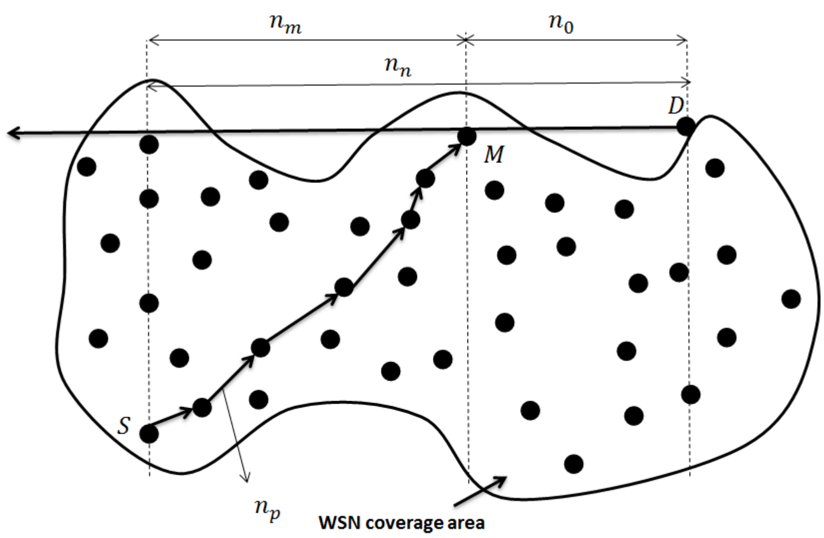

Figure 6.3: RREQ area

6.3 The impact of location errors on LAR and PRP performance

As specified in Section 5.1, the localisation process which takes place during WSN setup is inherently erroneous. Because of this nodes make use of estimated coordinates. It is therefore necessary to analyse the impact of the location errors on the routing protocols.

Both LAR and PRP use the same algorithm to find the shortest path from S to D. So it can be stated that for a like to like scenario, the effects of location errors on routing for LAR will be the same for PRP. However PRP uses LAR and PRA. PRA will calculate T_D so that D can save power and send hello packets when it is getting closer to its next neighbour. Because of location errors T_D is an estimated value. It is analysed further how location error influence PRP performances for the scenario presented in Figure 5.5. All the coordinates of A, B, C and E are estimated coordinates. Because of this, Eq. 5.7 can be rewritten as follows:

$$\hat{T}_D = \frac{\hat{d}}{v}, \tag{6.18}$$

where $\hat{d} = d \pm \epsilon$ is the estimated distance, d is the actual distance and ϵ is the location error which follows a normal distribution. Once $\hat{T}_D$ is estimated, D will know after how many seconds it will be in its next neighbour's communication range, so it will know when to start the broadcast of hello packets. However, because of the location error, D needs to start the broadcast of hello packets θ seconds sooner than the estimated $\hat{T_D}$, where θ is the confidence time interval influenced by the coordinates location errors ε. T_B is the time after which D starts the broadcast of hello packets when location error is considered and is calculated with the following equation:

$$T_B = \hat{T}_D - \theta. \tag{6.19}$$

For a general case, the location error which affects N's estimates of where and when D will rejoin the WSN through its next neighbour is presented in Figure 6.4. B is the point where D breaks the communication with its current neighbour N. C is the ideal point through which D will join M's communication area. The circle of center C and range ε represents the area where N estimates C position because of the location error. This entire circle is shifted to the right with $\triangle$ ($\triangle = \theta * v$) to compensate the location error and be able to increase to maximum the connectivity time between D and M. It can bee seen that:

- if C estimate, noted C_1, is within M communication range, by shifting to the right its coordinates with $\triangle$ it will make D to start sending hello packets much closer to C. However, in this case, because of the location error, the ideal communication time between D and M decreases. Because of this D will communicate with M for less time when compared with the ideal case. However, the time that D travels disconnected from WSN between B and C_1 increases which means that the data packets meant for D are sent to M's buffer.

Some particular cases may appear which need to be taken into account by D's neighbour before calculating T_B:

- **Case 1 -** this regards the scenario depicted in Figure 6.5 where $C(x_C, y_C)$ is the ideal point where D joins the network through node M. $C_1(x_{C1}, y_{C1})$ is C's estimate and $C_2(x_{C2}, y_{C2})$ is the point where D starts to send hello

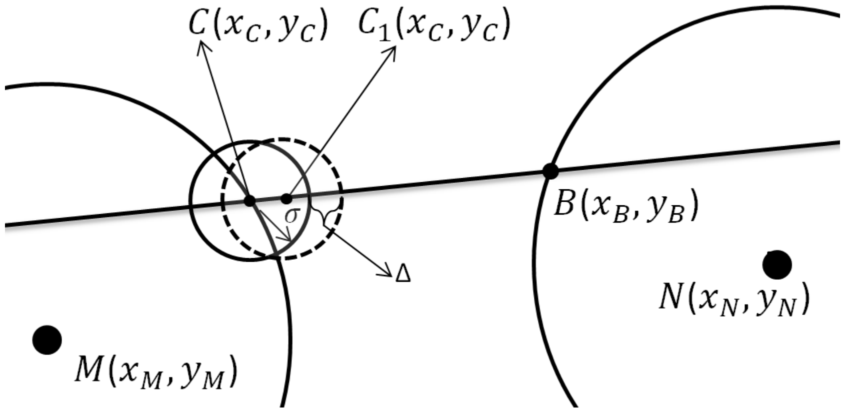

Figure 6.4: Location errors impact scenario

packets with θ seconds sooner. Because of the location error and the small distance that D travels disconnected between node N and M, D is informed to start sending hello packets when is still in N's communication range. To avoid this situation, N will compare the estimated distance between it and B (noted $\hat{d(N, B)}$) with the distance between it and C_2 (noted $\hat{d(N, C_2)}$). If $\hat{d(N, C_2)} < \hat{d(N, B)}$ it means that the distance that D travels between N and M disconnected is too small to delay sending hello packets and save power. In this case D will start sending hello packets as soon as it leaves N's communication range.

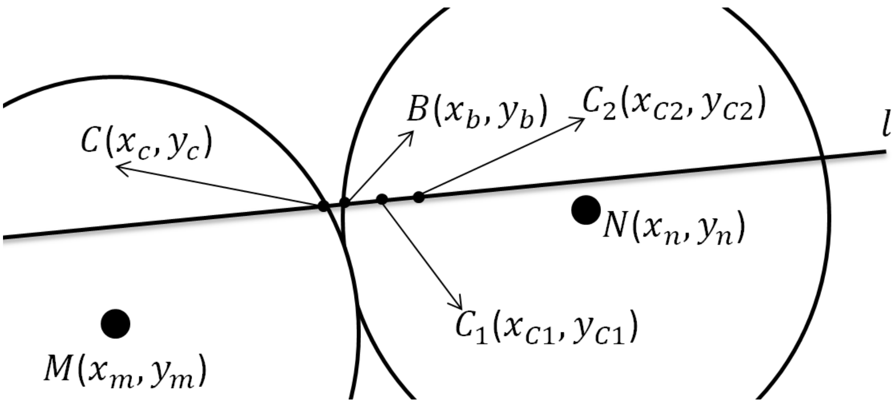

Figure 6.5: Location errors impact for an extreme scenario (when D's non-connectivity time is small)

- **Case 2** - this regards the scenario depicted in figure Figure 6.6 For this case it can bee seen that, because of the location error and short connectivity time between D and its next neighbour, D will send hello packets when it has already left M's communication range. In order to avoid this situation the coordinates of E (which were calculated in section 6.4) are also estimated. D is informed to start and send hello packets with θ_1 seconds sooner, ($\theta_1 = {}^{max(\varepsilon)}/_{v}$) if the distance between C and E is bigger than a certain threshold value which is chosen based on the maximum location error that can appear because of the position estimates ($d\,(\hat{C}, E) > max\,(\varepsilon)$).

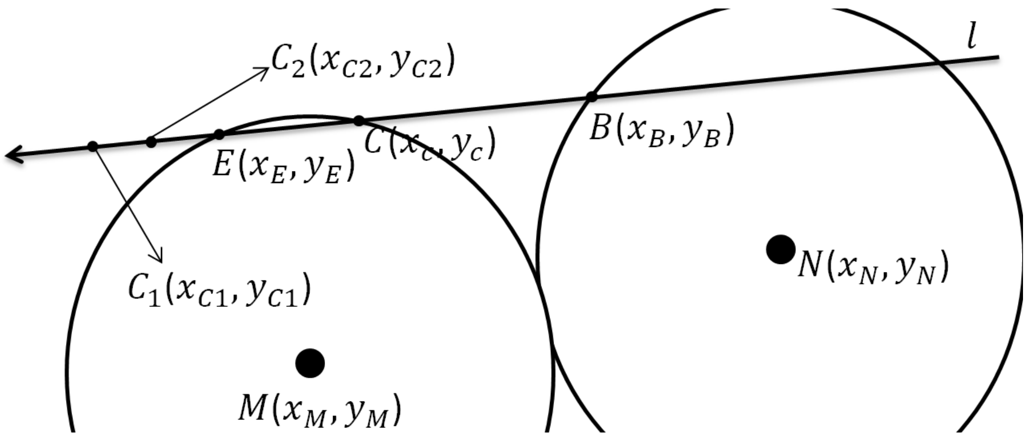

Figure 6.6: Location errors impact for an extreme scenario (when D's connectivity time is small)

For choosing θ a compromise needs to be found between the energy that is saved because of the time in which hello packets are not sent and the connectivity time that might be lost in the situation where D will be connected with its next neighbour for a short period of time. The value for θ can be chosen as follows:

$$\theta = \begin{cases} max\left(\varepsilon\right) & to\ maximize\ N_{PRP} \\ max\left(\varepsilon\right) - min\left(\varepsilon\right) & compromise\ between\ N_{PRP}\ and\ E_{PRP_{hello}} \\ min\left(\varepsilon\right) & to\ minimize\ E_{PRP_{hello}} \end{cases}$$

If T_{error} is the time lost from the connectivity due to location errors, for the case when D has energy constrains the equation for $Pckts_{PRP}$ becomes:

a) for the scenario when $T_{IC} < \tau_P + T_{ec} + T_{RB}$ (D to M connectivity is not sufficient for M to empty its buffer):

$$Pckts_{PRP_E} = \left[\frac{T_{IC} - \tau_p - T_{EC} - T_{error}}{t_p}\right]. \tag{6.20}$$

It can be seen that, for this scenario, $Pckts_{PRP}$ will not to be affected by location errors only if $T_{error} < \psi < t_p$ (calculus presented in Appendix 5).

b) for the case when $T_{IC} \geq \tau_P + T_{ec} + T_{RB}$ (D to M connectivity is sufficient for M to empty its buffer):

$$Pckts_{PRP_E} = 1 + \left[\frac{T_D + T_{error} + \tau_P - t_p * (n_p - 1)}{t_s}\right] + \left[\frac{T_{IC} - T_{error} - (\tau_P + T_{ec} + T_{RB})}{t_s}\right]. \tag{6.21}$$

For this case it can be seen that $Pckts_{PRP_E} \simeq Pckts_{PRP}$ for $(\forall)$ T_{error}.

So the condition which needs to be met by PRP to be resilient to location errors $(\forall)$ T_{error} is $T_{IC} \geq \tau_P + T_{ec} + T_{RB} + T_{error}$. The conclusions of this chapter are summarized in the next section.

6.4 Conclusion

In this chapter, the equations for the throughput, the total number of packets received by D ($Pckts$) and also the equations for the energy (E) were defined. It was seen that, when D travels disconnected from the WSN, LAR will not send any packets from S to D only if $T_{IC} < \tau_L + T_{RREP} + t_p * n_p$. $N_{LAR} = 0$ when D is connected all the time with the network only if $T_{IC} \leqq T_{RREQ} + T_{RREP} + t_p * n_p$. Also, $T_{wait} = 0$ will maximize $Pckts_{LAR}$ but it will also increase E_{LAR} so a compromise needs to be found when choosing T_{wait}.

For PRP, if $T_{IC} \geq \tau_P + T_{ec} + T_{RB}$ then $Pckts_{PRP}$ is described by Eq. 6.8, otherwise $Pckts_{PRP} = [{(T_{IC} - \tau_p - T_{ec})}/{t_p}]$. Based on the equations proposed for $Pckts_{LAR}$ and $Pckts_{PRP}$ it was analytically shown that $Pckts_{PRP} > Pckts_{LAR}$ for the scenario in which D travels disconnected from the WSN. Also, with the help of Eq. 6.17 it was analytically proven that the energy spent by LAR is bigger than the one spent by PRP when the same number of data packets are sent by both routing protocols. A discussion was made on how location errors can influence LAR and PRP performances.

The next chapter will describe the MATLAB simulator used to simulate a WSN scenario in which the S sends data to the D which follows the MMM. The LAR and PRP performance for different scenarios is presented further through MATLAB simulations.

7 MATLAB simulations and results

In this chapter MATLAB simulations are made to analyse the throughput (T), energy spent per received data packet packet (E) and end to end delay (ETED) for a realistic scenario in which a mobile sink (D), which follows the MMM, needs to receive information from a source (S) which resides in the WSN. T and E are the ones defined in Chapter 6. The ETED represents the time spent per received data packet and is calculated by dividing the simulation time with the total number of data packets received at D. In section 7.1 the MATLAB simulator is described. The simulation scenario is presented in section 7.2 together with the first results obtained when LAR, AODV, DSR and PRP are used for routing and D does not have energy constraints. In section 7.3 simulations are made for the scenario where D has energy constraints and different parameters are varied like: the connectivity of D to the, WSN density, D's speed and WSN size. The obtained results are discussed after each section and summarized at the end.

7.1 The WSN MATLAB simulator

To be able to compare the performance of routing protocols recommended for MANETs in a WSN, the MATLAB simulator presented in [128], [129] and [130] is used for WSN setup which allows: the modification of the network topology; the use of accurate or erroneous localisation; the possibility to adjust energy-consumption related parameters and to imitate the behaviour of a realistic transmission channel. The LAR, AODV, DSR and PRP routing protocols functions were developed and added to this WSN simulator. This sub-chapter presents the general features of the developed simulator.

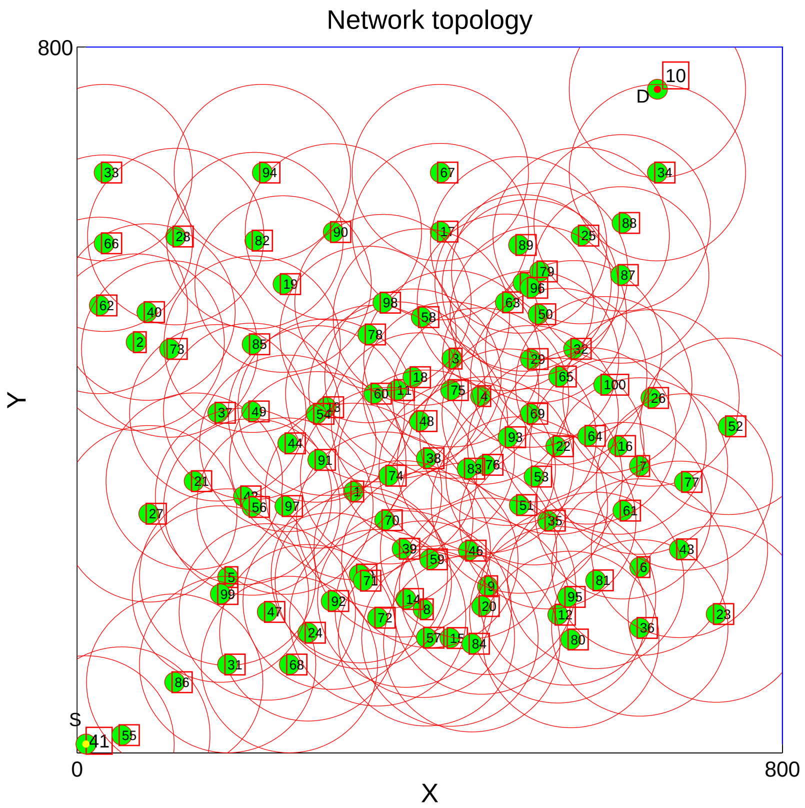

Figure 7.1: WSN Simulation Scenario

The Physical (PHY) and the Network Layer are the ones considered for the MATLAB simulator. The focus of this research was on the routing aspect of the network functionality so other layers such as the MAC, Transport or the Application Layer, were not considered. The simulator makes use of a main MATLAB function and of multiple scripts with globally and locally defined variables. The following steps are taken for each simulation:

- WSN setup - the parameters which define the scenario are initiated at the beginning such as: network area width and length, number of nodes, number of packets, D's speed, node transmission range, node transmission power, path loss exponent, the standard deviation for shadowing model, sensitivity threshold, route request packet size, route reply packet size, hello packet size, data packet size, data rate, energy per bit spent on transmission, energy per bit spent for reception, initial node energy. The node placement can be considered known and fixed or random following a normal distribution.

- The simulator is based on the fact that each transmitting node follows the same steps as in the following description: the source (S) needs to send data packets to a certain destination (D); each node (in the first case the S) checks which nodes from its transmission range (R) are the best candidates to receive and forwards the data towards D; based on the adopted forwarding strategy the best candidate is selected and the data packets are sent.

Simulator Parameters (unit)	Value
Network area width (m)	800
Network area length (m)	800
Number of nodes	100
D speed (km/h)	40
Transmission range (m)	100
Transmit power (W) [131]	600e-3
Path loss exponent [132]	4
Standard deviation for shadowing model [132]	5.6
Sensitivity threshold (dBm) [133]	-81
Route request packet size (b) [134]	104
Data packet size (b) [[134]]	104
Data rate (kB/s)	250
Energy per bit spent on transmission (J/b) [[135]]	2.5e-07
Energy per bit spent for reception (J/b) [[135]]	1.5e-0.7
Initial node energy (J) [[133]]	1

Table 7.1: MATLAB Simulation Parameters

The MATLAB WSN simulator presented in [128], [129] and [130] is used only to setup a WSN scenario in which S sends packets to D. LAR, AODV, DSR and PRP routing protocols are implemented and added to this WSN simulator. These are used to send data packets from the S to D which follows the MMM. The node placement follows a normal distribution as depicted in Figure 7.1 where the number of nodes is set to 100 and the area size to 800 m in length and width. Node 41 is the S and node 10 is the D. At time $t = 0$ S finds a path to D and starts sending data packets. During this time, D follows the MMM moving in a straight line with a constant speed of 40 km/h from right to left. It is understood that, during its movement, D will exit and enter the WSN coverage area. Different simulations are made as D to WSN connectivity varies between 10% and 100% (with a 5% increment) and LAR, AODV, DSR and PRP are used. The parameters used for simulation are presented in Table 7.1. A single S and D are used for this simulation in order to have a simplified scenario. In sub-chapter 5.3.1 it is stated that the MAC affects LAR performance. However, it can be assumed that the MAC layer impact on the overall performance remains the same for all the simulation parameter variations because LAR, AODV, DSR and PRP are compared in a like-for-like simulation platform. So the MAC layer is not considered in the following simulations. It is described next how LAR, DSR, AODV and PRP were implemented.

7.1.1 LAR and PRP routing diagrams

LAR is described in [37] and presented in section 4.2. For a better understanding about how this protocol was implemented in MATLAB the routing diagram presented in Figure 7.2 is used and explained as follows:

- In the beginning the WSN is formed through the block “WSN Setup” where the parameters presented earlier are defined.

- The next step is to check if a path to D is known. If it is known then a data packet is sent by S to D. If all the packets are sent (“Packets=0” is true) the simulation ends. If there are still packets to send S will check to see if the path is still valid. When D leaves the WSN coverage area, its neighbour will inform S that the path is broken, so S needs to check the status of the path each time before sending a data packet.
- If the path to D is not known S will need to send RREQ packets in the network to search for a path. To avoid the situation in which S sends a high number of RREQ packets because D will rejoin the WSN after a long period of time, a fixed number of RREQs packets (No_RREQ=limit) are sent after which S will pause the RREQ transmission for a certain period of time. “No_RREQ” is used to count the number of sent RREQ packets.
- LAR uses route request zones instead of flooding the entire WSN with RREQs packets. Before sending RREQs packets S will check if D’s previous coordinates are known. If these are known then an expected zone is defined based on S’s coordinates, D’s previous coordinates and D’s speed and time passed since D was connected with the WSN. If D’s previous coordinates are not known, S will send RREQ packets in the entire WSN.
- Once the expectation zone is defined S will start sending RREQ packets. When a path is found (always the shortest path is chosen), “No_RREQ” is set to 0 and S will start sending data packets to D.
- For a better understanding about how the LAR simulator was implemented, the LAR pseudo-code together with its description is presented in Appendix 1.6.

The PRP was described in chapter 5 and the block diagram presented in Figure 7.3 is used to better understand how the simulator was implemented. This is described

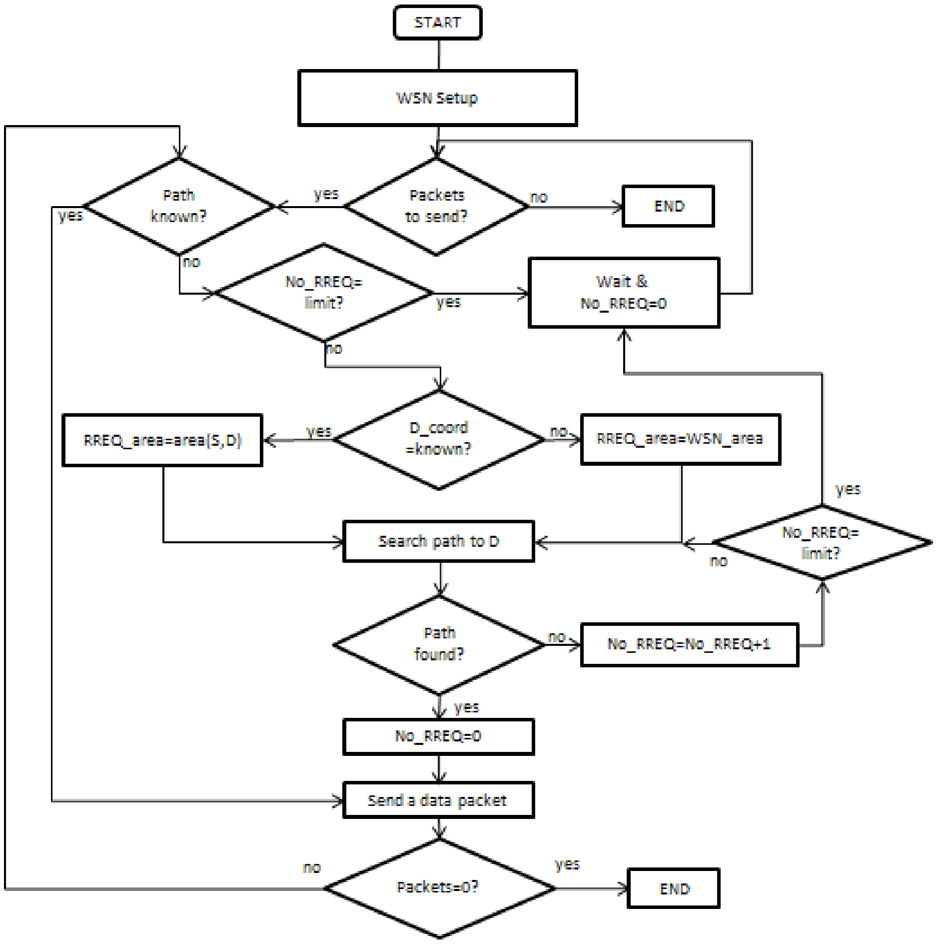

Figure 7.2: LAR routing diagram

as follows:

- In the beginning the WSN is formed through the block "WSN Setup" where the parameters presented earlier are defined.

- The next step is to check if a path to D is known. If a path is known S needs to check if this path is to D or D's next neighbour. If the path is to D ("buffer = false") then data packets are sent to D until all the packets are sent or D will leave the WSN and S will be informed by D's neighbour about this. When D will break the communication, (buffer=true and path_known=false) S will search a path to D's next neighbour by sending RREQ packets in the RREQ area defined by S and D's next neighbour coordinates. When a path to D's next neighbour is found (buffer=true and path_found=true) data packets are sent to D's next neighbour. When D rejoins the WSN S will be informed to stop sending data packets until all the buffered packets are received by D ("buffer" in this case becomes false). However, if D does not rejoin the WSN after an estimated time (T_D seconds which is the deconnectivity time that D is estimated to travel before rejoining the WSN), S will stop sending data packets to D's next neighbour and it will start to send RREQ packets in the entire WSN to search a path do D.

- To avoid the situation were RREQs are sent in the area through which D will not rejoin the WSN (because D changes direction of movement or it stops moving and it will remain outside the WSN), a certain number of RREQs are sent in the network after which S will pause the search for a path to D. The estimated area in which RREQs are sent will become the entire WSN area. Due to space limitation in Figure 7.3, the block diagrams for this process (which need to be added when "path found" is false) are not presented in PRP block diagram, however these are similar with the ones presented in LAR diagram.
- For a better understanding about how the PRP simulator was implemented, the PRP pseudo-code together with its description is presented in Appendix 1.7.

DSR is described in [38] and presented in section 4.2. When compared with LAR, DSR saves five routes from S to D and not just one. When the communication between D and the WSN breaks S will test the rest of four routes saved in cache and after this it will send RREQs in the entire WSN to search for a new path. AODV is described in [39] and presented in section 4.2. When D exits the WSN area it starts to broadcast hello messages which are used to detect the out-of-connectivity-range neighbours. S will send RREQs packets in the entire WSN to search for a path to D.

MATLAB simulations are made next for a scenario in which D has no energy constraints and LAR, AODV, DSR and PRP are used for routing.

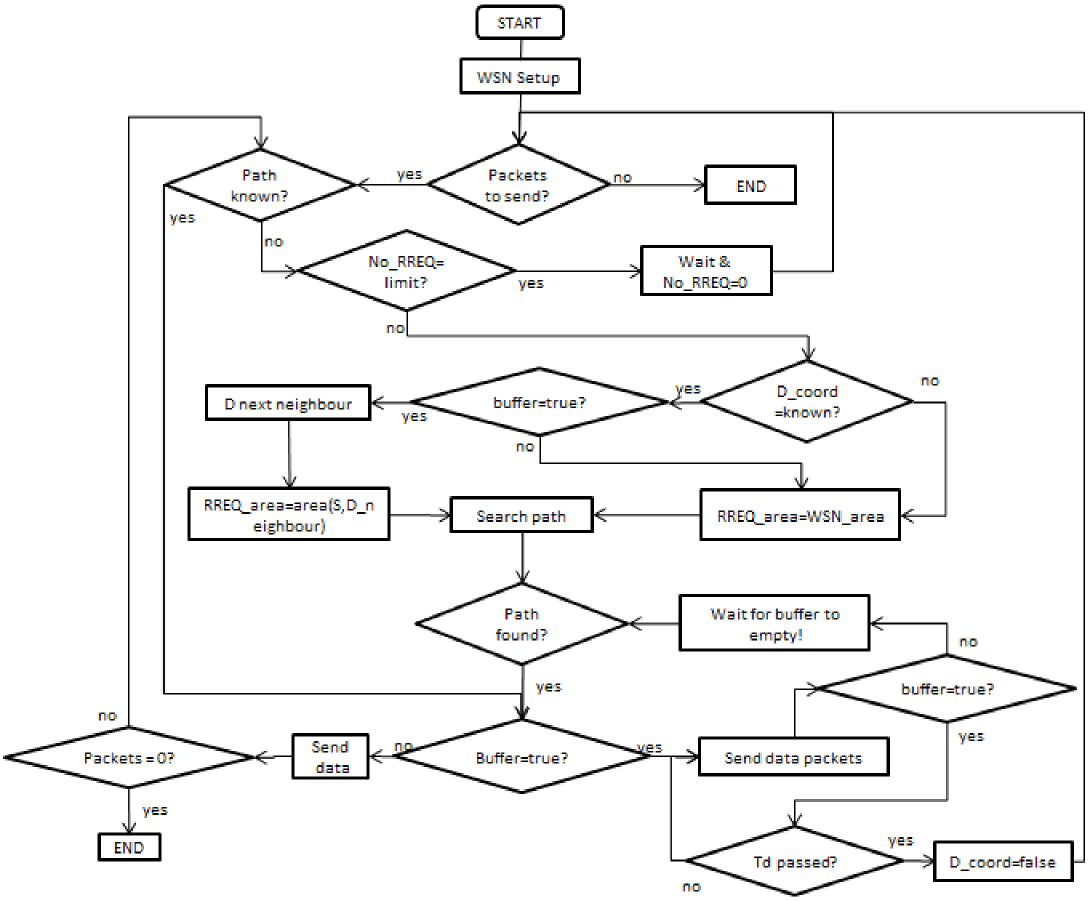

Figure 7.3: PRP routing diagram

7.2 Simulations and results for D with no energy constraints

For the next simulations it is assumed that D has no energy constraints. This means that as soon as D leaves its neighbour coverage area, it starts to send hello packets to find its next neighbour (it will use a small waiting time between two hello packets). By doing this it will join its next neighbour exactly when it enters its communication range. First, it will be analysed how t_s (time between sending two data packets) can influence the T for LAR and PRP (section 7.2.1). In section 7.2.2 simulations are made where LAR, AODV, DSR and PRP are used and the D to WSN connectivity varies between 10% and 100%. In section 7.2.3 the time between two RREQs is increased to reduce the E.

7.2.1 t_s impact on LAR and PRP performance

A simulation is made in which t_s is varied from 0.0024 seconds to 0.1 seconds using a 0.0008 seconds increment. The increment was chosen based on the time needed for a data packet to be sent between two nodes which are one hop away. Also, 0.0024 seconds represents the minimum time that S needs to wait before sending a data packet to avoid interferences from the previous sent one (time needed for a data packet to be sent three hops away from S). For each t_s the D to WSN connectivity is varied between 10% and 95% using a 5% increment so 18 different simulation scenarios are considered. The obtained T for PRP and LAR is presented in Figure 7.4. The T variation comparison between PRP and LAR is presented in Figure 7.5. It can be seen that PRP outperforms LAR and that the difference between these protocols is rising to a constant value of 73%. For PRP when t_s is bellow 0.04 seconds, the data packets sent to D's next neighbour buffer is bigger than the ones which can be sent from this buffer to D. This means that for low connectivity scenarios there is no sufficient time to empty the buffer when D reconnects with the WSN. Because of this, the PRP T improvement over LAR is not constant for t_s values bellow 0.04 seconds (as it was discussed in Section 7.1.2 and Section 7.3) . For $t_s > 0.04$, D has time to receive all the buffered packets from its next neighbour so PRP will always send 73% more data packets when compared with LAR.

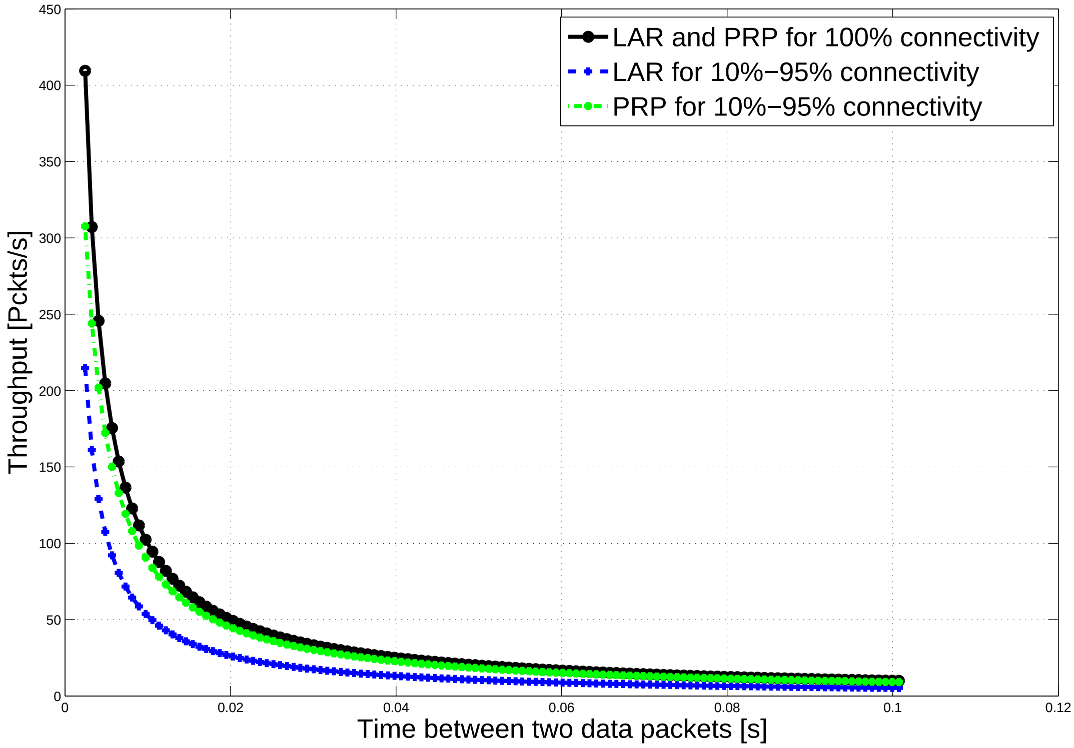

Figure 7.4: T when t_s is varied

7.2.2 LAR, AODV, DSR and PRP simulation results

In [9], [68], [136], [137], and [138], the source generates data from 56 bytes to 250 bytes per second. For the next simulations it will be considered that S will send 250 bytes per second which means that it generates one data packet each 0.05 seconds, so it will send 20 packets per second. It was seen that, for 100% D to WSN connectivity, LAR, AODV and PRP send a maximum of 19.2 packets per second and DSR 19. Figure 7.6 shows the T obtained by each routing protocol as the D to WSN connectivity is varied between 10% and 95% with a 5% increment. PRP uses LAR to send the data packets from S to D for the 100% connectivity scenario. When the

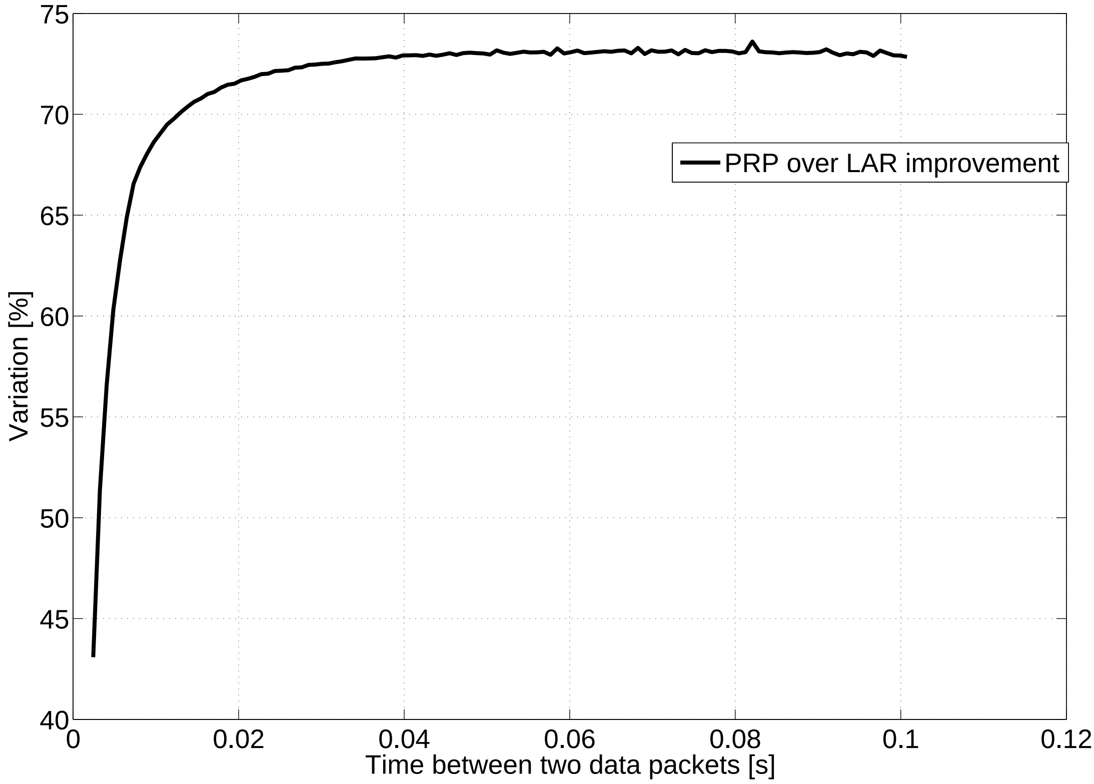

Figure 7.5: T variation when t_s is varied

communication between these two nodes is broken, because D moves outside the network range, S is informed which node will be D's next neighbour. S will find a path and send data packets to D's next neighbour. D's next neighbour will save these packets in its buffer until D rejoins the WSN. Overall, across the 10%-95% connectivity scenario, PRP obtains an average T of 17.48 packets per second, LAR and AODV of 10.1 packets per second and DSR of 10.05. DSR has the worst results because it gathers information about five different routes from S to D. When the communication between D and the WSN breaks, S will first test each of these five routes. Only afterwords it will start to search a new path to D. The time lost on testing the validity of the five routes affects DSR performance, decreasing the number of sent packets.

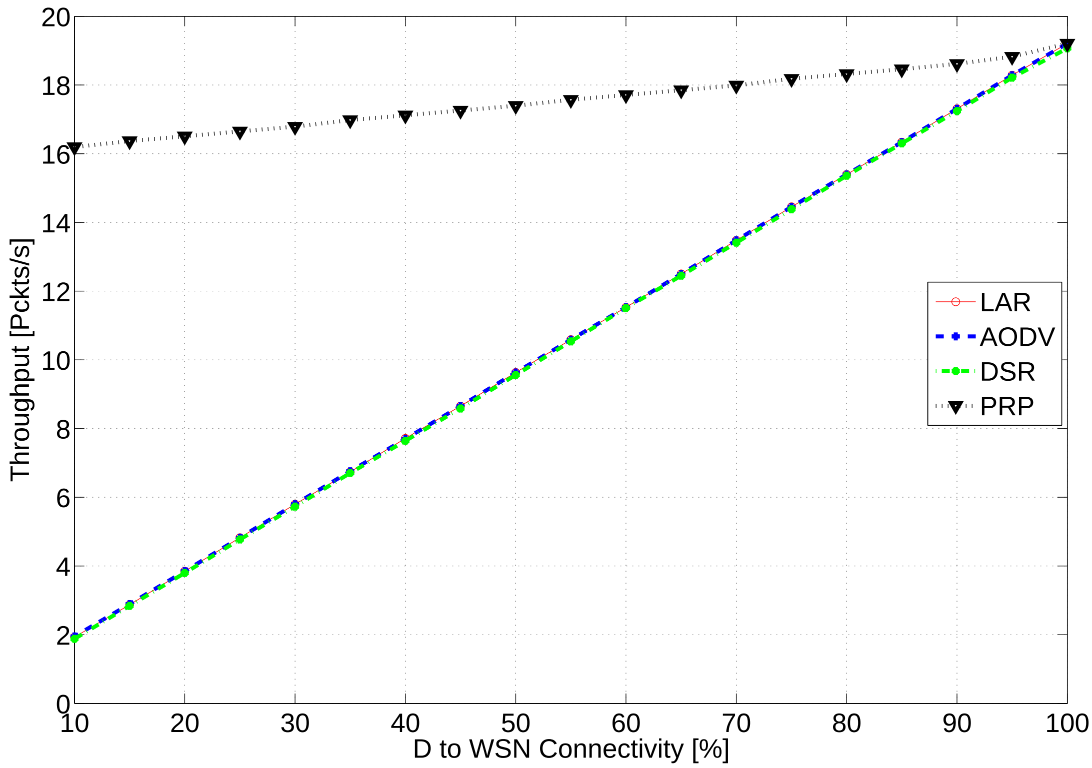

Figure 7.6: T when D to WSN connectivity varies

Figure 7.7 depicts the obtained ETED. It can be seen that PRP outperforms all the other routing protocols because it has the highest T. This is possible because once D re-joins the WSN it will receive data packets from its neighbour's buffer. The results obtained for 100% D to WSN connectivity and the average for 10% to 95% are summarized in Table 7.2.2 . Overall, for the 10% to 95% connectivity scenario, PRP has on average a 9.6% end to end delay increase when compared with the best case scenario where D to WSN connectivity is 100% (for 10% connectivity the ETED increases with 18.6%). For LAR and AODV the average end to end delay (AETED) is increased with 88.4%, and for DSR with 89.6%.

The E for each protocol is presented in Figure 7.8 and summarized in Table 7.3. E is influenced by the energy spent on RREQs and routing. LAR uses zone requests

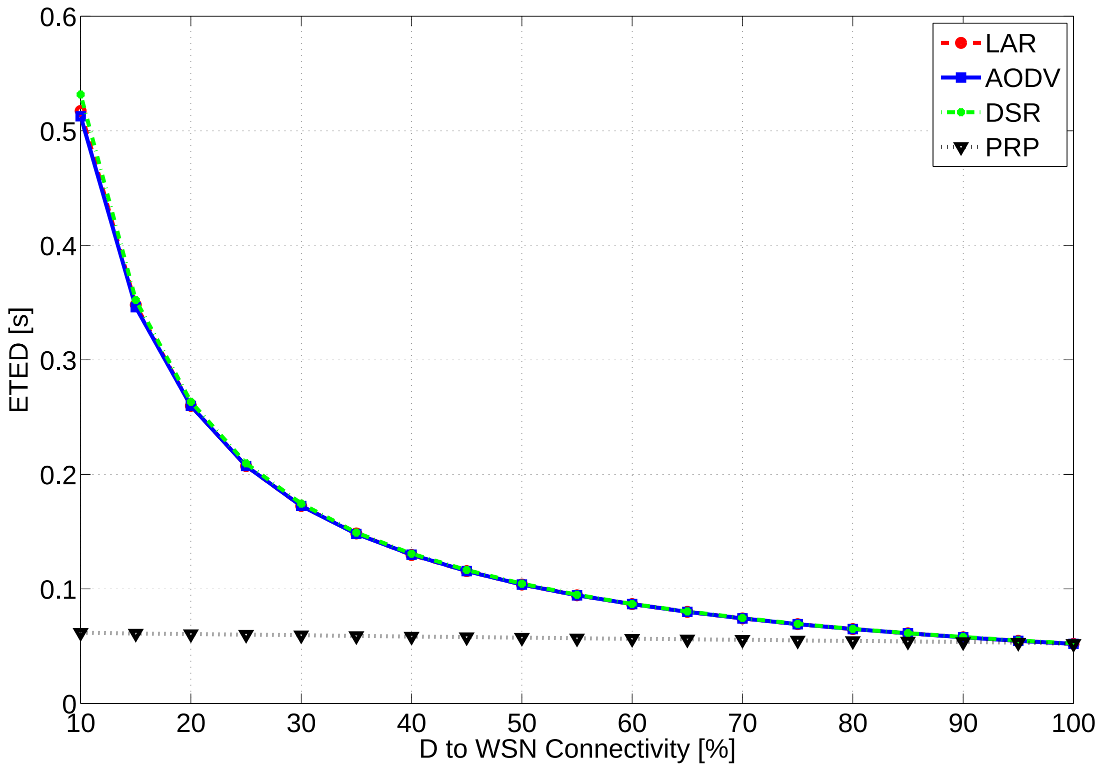

Figure 7.7: ETED when D to WSN connectivity varies

calculated based on S and D coordinates to establish an area in which to send RREQ packets. In time, these RREQ zones become smaller when D moves from right to left so less nodes are flooded with RREQ packets. Both AODV and DSR flood the entire WSN with RREQs when D breaks the communication. In addition, DSR will check five more routes from S to D which were previously saved when a path from S to D was searched. It can be seen that PRP maintains its energy consumption of 0.0006 J per packet for all the D to WSN connectivity scenarios. On average, for the 10% to 95% connectivity scenarios, AODV and DSR use with 33% and 30% more energy when compared with LAR.

D-WSN con.	LAR		AODV		DSR		PRP	
	T	AETED	T	AETED	T	AETED	T	AETED
100%	19.2	0.052s	19.2	0.052s	19.06	0.0524s	19.2	0.052s
10%-95%	10.1	0.098s	10.1	0.098s	10.05	0.099s	17.48	0.057s

Table 7.2: T and AETED when D to WSN connectivity varies

D-WSN con.	LAR	AODV	DSR	PRP
	E(J)	E(J)	E(J)	E(J)
100%	0.0006	0.00064	0.00066	0.0006
10%-95%	0.0539	0.0718	0.0705	0.0006

Table 7.3: E when D to WSN connectivity varies

7.2.3 $\mathbf{T_{RREQ}}$ impact on LAR performance

The energy spent on RREQs is directly influenced by the number of RREQ packets sent in the WSN by S to search a path to D (by the waiting time between two RREQs noted in the previous chapter with T_{RREQ}). However, although increasing the time between two RREQs will reduce E, it will also reduce the T. This is due to the time spent by S to find a path to D (which is also directly influenced by the waiting time between two RREQs). In Figure 7.9 LAR E and T variations when the time between two RREQs increases from $t_0 = 0.002$s (the time that it takes for a RREQ packet to reach the furthest node in the WSN from S and return) to $50 * t_0$ are presented. Each simulation is the average of 18 different scenarios in which D to WSN connectivity increases from 10% to 95% using a 5% increment. For t_0 and 10% to 95% connectivity, LAR has a T of 10.1 packets per second and uses 0.0539 J per packet to do this. We denote this as LAR1. For $50 * t_0$ LAR has a T of 9.62 packets per second and the E is reduced to 0.0019J. We denote this as LAR2.

For 10% to 95% connectivity, LAR T drops from 10.1 to 9.62 packets per second which represents a 4.8% decrease. Also the E decreases from 0.0539J to 0.0019J, which represents a 96.4% decreases. The waiting time between two route requests

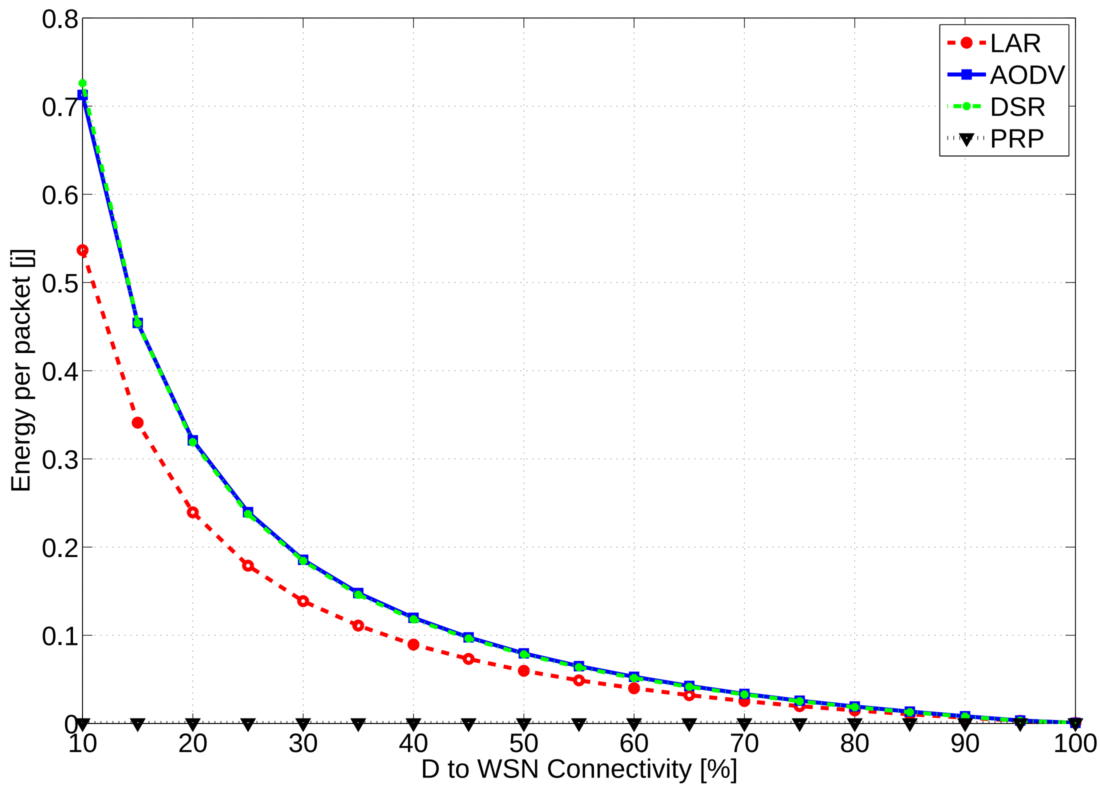

Figure 7.8: E when D to WSN connectivity varies

needs to be chosen to have a good compromise between the T drop and E improvement. It can be seen that for $12 * t_0$, the T decreases by 0.8% and the E by 90.33%. A maximum 1% T drop and over 90% E saving will be considered to be a good compromise when choosing the waiting time between two route request. This scenario for $12 * t_0$ will be noted as LAR3. The overall results regarding the T, AETED and E for PRP, LAR2 and LAR3 are summarized in Table 7.2.3. In Figure 7.12 and Figure 7.13, LAR1, LAR2 and LAR3 performance in terms of T and E are compared with PRP.

When the time between two RREQ increases the D to WSN connectivity time decreases. However, this is not true for all the scenarios in which D to WSN connectivity varies from 10% to 95%. Because of the network topology the situation

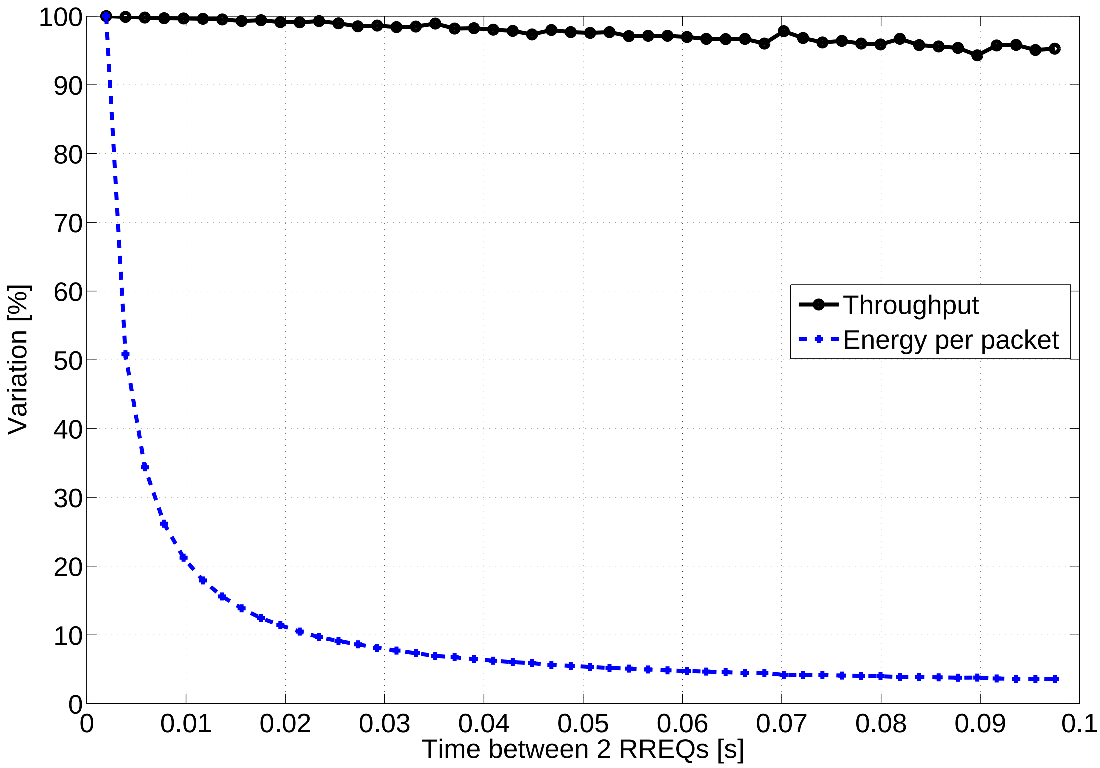

Figure 7.9: T and E variation for LAR when the time between two RREQs is varied

presented in Figure 7.10 and Figure 7.11 explains why$T$ does not have a uniform evolution. It can be seen that a bigger time between two RREQs does not necessarily mean a smaller connectivity time between D and the WSN.

From the above simulations the following conclusions can be drawn:

- For $t_s > 0.04$ PRP has a 73% T improvement over LAR.

- When D to WSN connectivity varies between 10% and 95%, PRP has a T improvement of 73% over LAR and AODV and of 74% over DSR. AODV and DSR use with 33% more energy and LAR with 30% when compared with PRP. Also, when compared with the 100% connectivity scenario, PRP has a 9.6% AETED increase, LAR and AODV of 88.4% and DSR of 89.6%.

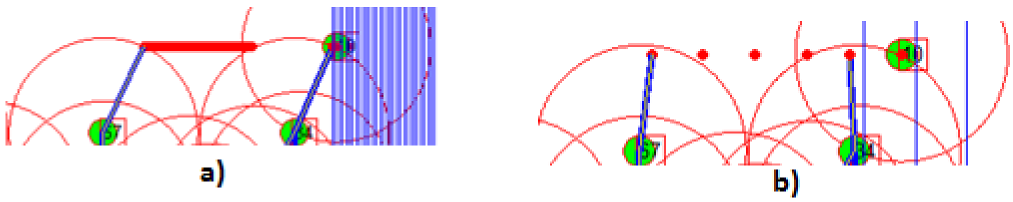

Figure 7.10: A path to D is found when the time between two RREQs is: a) 0.002 s; b) 0.04 s

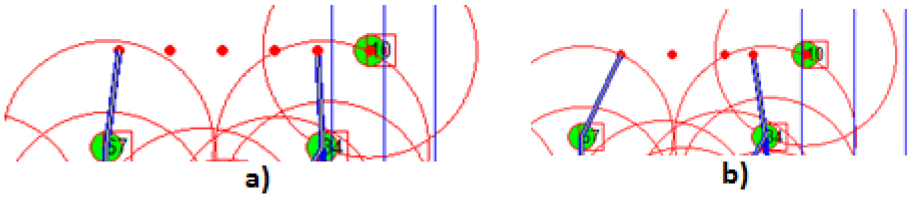

Figure 7.11: A path to D is found when the time between two RREQs is: a) 0.04s; b) 0.06s

- When T_{RREQ} increases to $50 * t_0$, LAR E is reduced with 96.4% but its T also decreases with 4.8%. A good compromise is considered to be LAR3 (for $T_{RREQ} = 12 * t_0$) which uses with 90.33% less E and has a T drop of 0.8%. This is considered to be a good compromise so LAR3 will be considered for further simulations.

In the next section, simulations are made for the case when D has energy constraints.

7.3 Simulations for D with energy constraints

For all the above simulations D was considered not to have energy constraints. This assumption affects only PRP performance because D starts to send hello packets to rejoin the WSN as soon as it breaks the communication with its neighbour. In this way, the communication time between D and its next neighbour is maximized and D will receive the maximum number of packets from S for this scenario. This will be noted as PRP1. If D has energy constraints it will need to send hello packets

D con.	PRP			LAR2			LAR3		
	T	AETED	E(J)	T	AETED	E(J)	T	AETED	E(J)
100%	19.2	0.052s	0.0006	19.2	0.052s	0.0006	19.2	0.052s	0.0006
10%-95%	17.48	0.057s	0.0006	9.62	0.1038s	0.0019	10.03	0.099s	0.0052

Table 7.4: T, AETED and E for PRP, LAR2 and LAR3 when D to WSN connectivity varies

only when it is getting closer to its next neighbour location in order to save power (the time spent between two hello packets is considered to be T_{hello}). This is possible because D's neighbour approximates when and where D will rejoin the WSN as it was presented in Chapters 5.2 and 5.3. Location error estimates (ϵ) will influence PRP performance in terms of:

- **Reducing the T**- this happens when $\epsilon < 0$ which implies that D will start to send hello packets to rejoin the WSN after it entered its next neighbour coverage area. This means that the connectivity time between D and its next neighbour will decrease which means that the T will decrease when compared with PRP1 only when there is no sufficient time for D's neighbour to send all the packets from its buffer ($T_{IC} < \tau_P + T_{ec} + T_{RB}$ as it was shown in the previous chapter). To compensate for the connectivity time lost due to location errors, D will start sending hello packets with θ seconds sooner (Δ is the distance associated with θ). If $\epsilon > 0$ D will start to send hello packets when it is getting closer to its next neighbour coverage area. For this case the T will not be affected.

- **Increasing** E_{hello} - this happens when $\epsilon > 0$ which implies that the total number of hello packets sent by D when it needs to re-join the WSN will increase. For this case, in order to minimize E_{hello}, the number of hello packets which are sent by D needs to be minimized so an optimal value for T_{hello} needs to be considered.

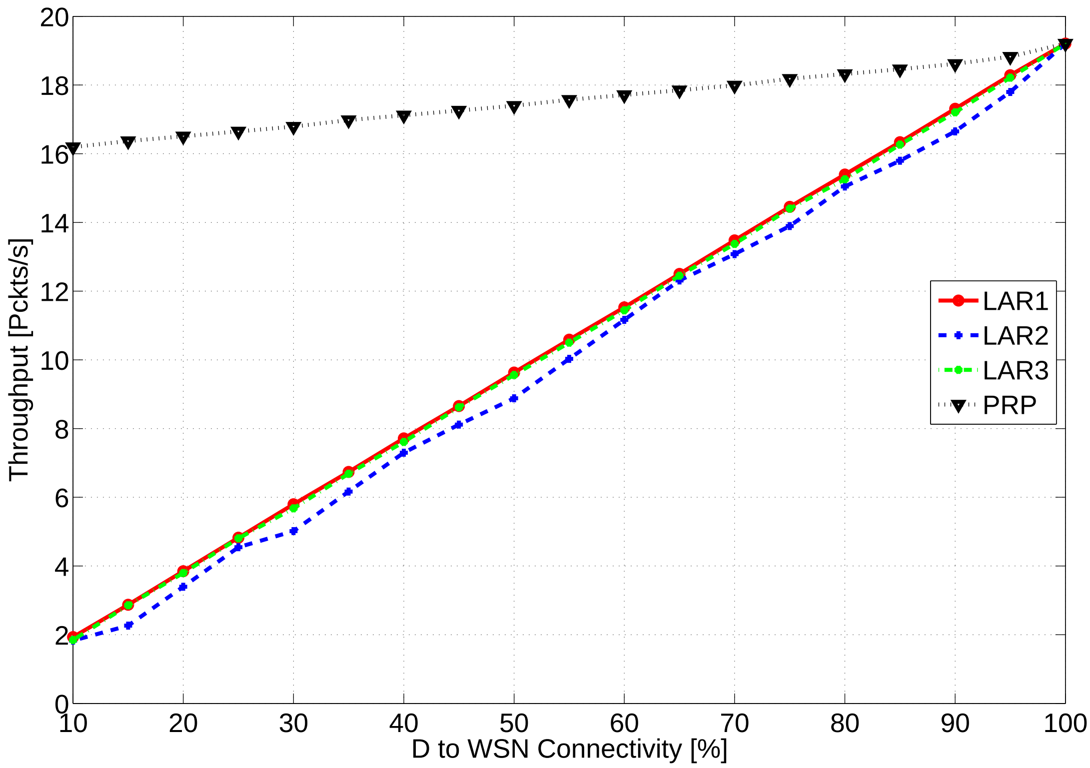

Figure 7.12: T when D to WSN connectivity varies

The ideal values for $\triangle$ and T_{hello} are those which will increase D to WSN connectivity to a value equal or grater than $\tau_P + T_{ec} + T_{RB}$ (time needed to send to D all the buffered packets) and will keep the E to a minimum (D will send a maximum of two hello packets). The choice of the two values is explained in the following paragraphs.

7.3.1 Choosing the optimal $\triangle$ to maximize the T

PRP will start to send hello packets θ seconds sooner ($\theta = {}^{\Delta}/_{v}$) than its neighbour estimates to compensate for the ϵ thus to maximize the connectivity between D and WSN. The next simulations are made to see how ϵ influence PRP performance and which values are needed for Δ to obtain the best T. It is considered that ϵ follows a normal distribution with a standard deviation σ of 30% from R as it was described

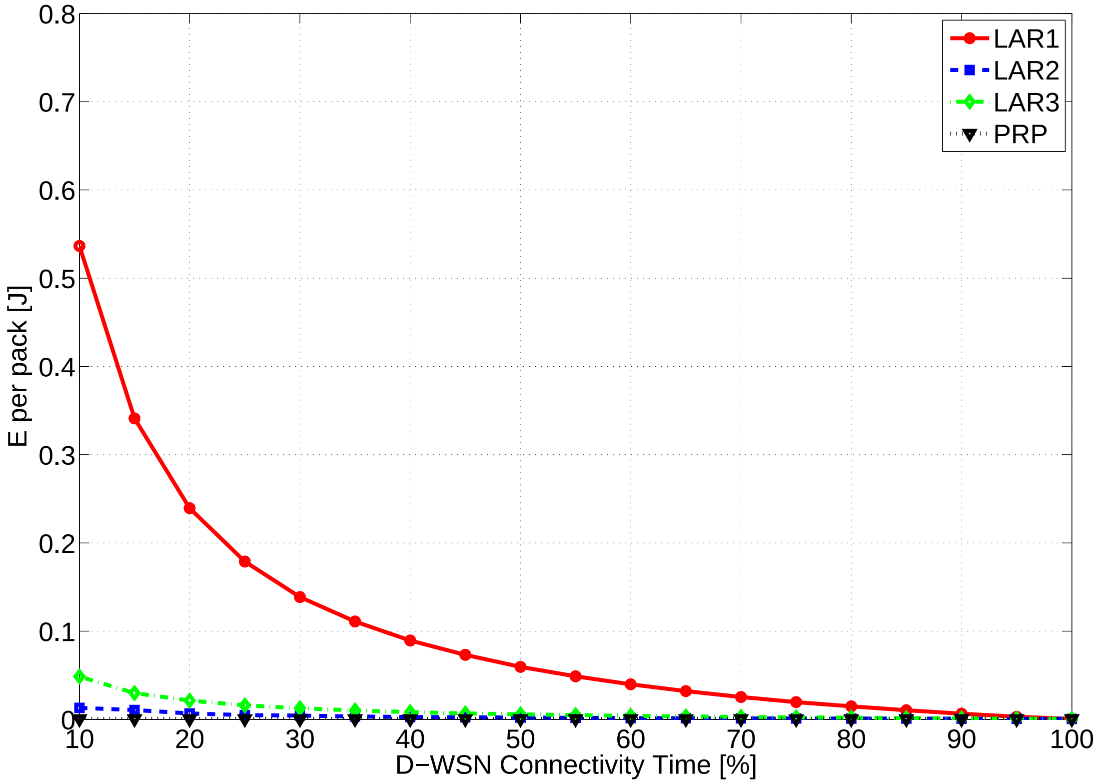

Figure 7.13: E variation when D to WSN connectivity varies

in [130] (for our simulations R is 100 m). As the error model is Gaussian, the error itself can vary between $(-3\sigma, 3\sigma)$ which is equal to a range of $(-90m, 90m)$.

Figure 7.14 represents the T and E variation in percentage (when compared with PRP1) when T_{hello} is fixed and Δ varies between $(0m, 90m)$(the equivalent being $(0,\ 3\sigma)$). T_{hello} is chosen to be $10 * t_p$ were t_p is the time needed to send a hello packet one hop away. It can be seen that when $\triangle$ increases (which means that θ also increases), the T increases reaching a value equal with PRP1 for $\Delta > 60m$. However, the E will drop between $(0m,\ 20m)$ after which it will start to rise again. This fluctuation appears because a large value for Δ will add more distance then needed to compensate for ϵ which will lead to to a longer time during which hello

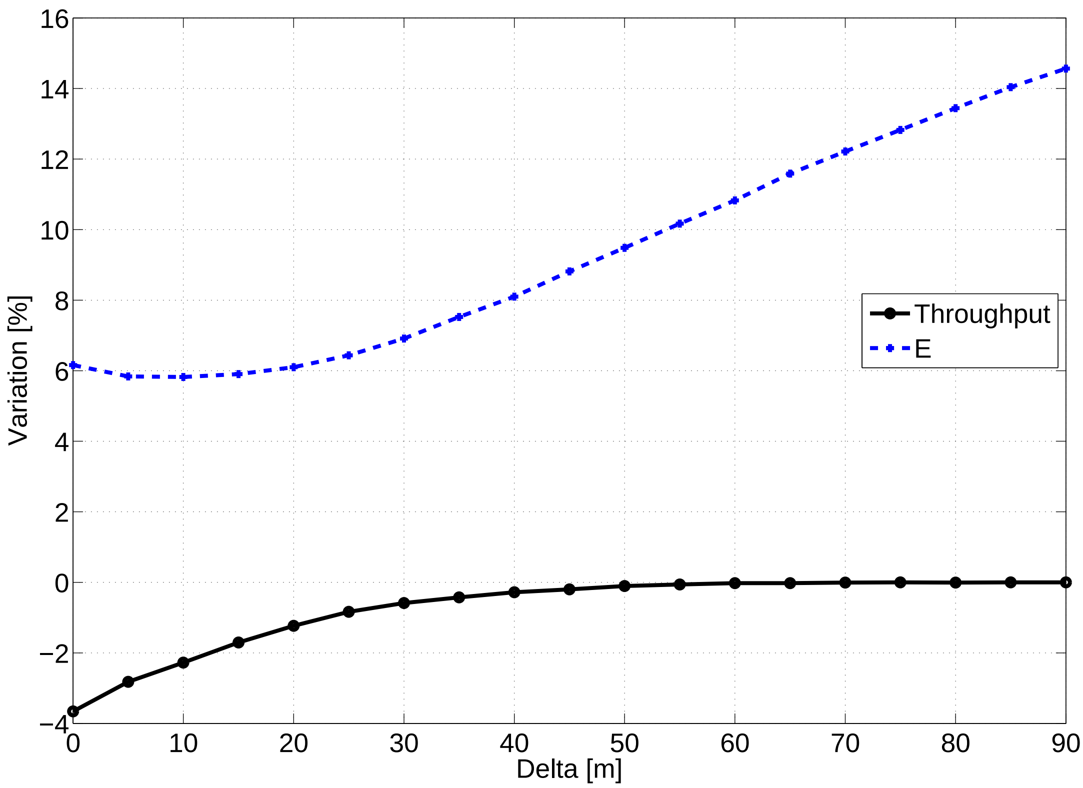

Figure 7.14: T and E variation for PRP when Δ is varied and T_{hello} is fixed

packets are sent until it will join the WSN, thus more energy spent on hello packets. It can be seen that choosing $\triangle$ will lead to a trade-off between E and T. However, E can be reduced further if T_{hello} is increased. Choosing a big value for Δ and an ideal value for T_{hello} will lead to a high T and a small E. It is discussed next how T_{hello} is chosen.

7.3.2 Choosing the optimal $\mathrm{T_{hello}}$ to minimize E

The scenario presented in Figure 7.15 is used to explain how the best value for T_{hello} can be estimated in order to save power and maintain a maximum T. The following notations are used: B is the point where D breaks communication with node N;

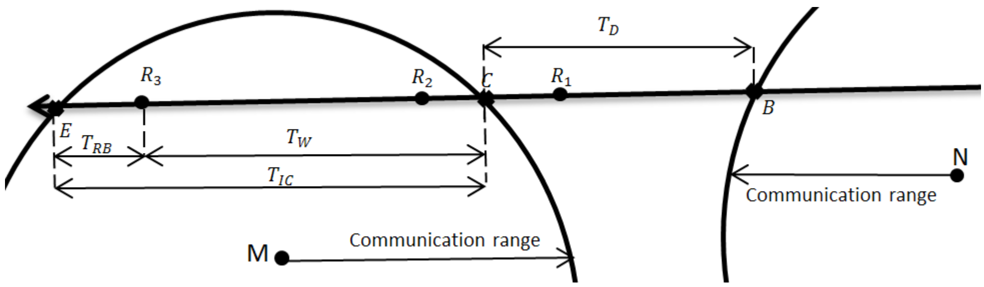

Figure 7.15: Scenario showing when D joins the WSN

M is D's next neighbour; C is the location where D will start to send hello packets for the ideal case (if N estimates without error when D will join M communication range); T_{RB} (time to receive from buffer) is the time needed by D to receive all M's buffered packets; T_W (time window) represents the maximum time that D can loose from T_{IC} (ideal connectivity time) and still have time to receive all M's buffered packets. Due to location errors D will start to send hello packets between R_1 and R_2 ($R_1 = C - 3\sigma$; $R_2 = C + 3\sigma$). However, if $T_{IC} \geq T_{RB}$ then the T is not affected so R_3 is the location up to which D would need to start its communication with M to obtain the best T. If $\hat{C}$ (C's estimate) is between C and R_3 then $T_{IC} \geq T_{RB}$ which means that the T will not be affected by the location errors. If $\hat{C}$ is between C and B, D will start to send hello packets outside the WSN each T_{hello} seconds. For an ideal case (where the energy spent on hello packets is minimum and the T is maximum) D will send one hello packet once its position is between C and B and the second hello packet when its position is between C and R_3. This can be achievable if D will receive information about the values of T_W (time window) and T_{RB} (time to receive from buffer) from its ex-neighbour. These can be calculated using Eq.7.1 and Eq.7.2:

$$T_{RB} = \frac{T_W + T_D}{t_s} * t_p \Leftrightarrow T_{RB} = \frac{T_{IC} - T_{RB} + T_D}{t_s} * t_p \Leftrightarrow T_{RB} = \frac{T_{IC} + T_D}{t_s + t_p} * t_p; \quad (7.1)$$

$$T_W = T_{IC} - T_{RB} \Leftrightarrow T_W = T_{IC} - \frac{T_{IC} + T_D}{t_s + t_p} * t_p; \quad (7.2)$$

where T_{IC} is estimated based on C and E coordinates, t_s is the time spent by S between sending two data packets and t_p is the time that it takes for a packet to be sent from M to D.

By knowing an estimate of T_W and an estimate where D will need to start and send hello packets, D can estimate which T_{hello} to use in order to send a minimum number of hello packets (minimize the energy spent on hello packets) but keep a maximum T. If $T_{hello} = T_W$ the following two scenarios may take place:

- If ϵ is between $(0, 3\sigma)$ (first hello packet is sent between C and B) then the total number of hello packets sent is equal to two thus the energy spent on hello packets is minimum. Also the location where D will send the second hello packet will be between C and R_3 which means that the T is maximized.

- If ϵ is between $(-3\sigma, 0)$ then D will send only one hello packet between C and R_3 and it will start to receive packets from its neighbour at that point.

The following two extreme scenarios can also take place due to location errors because B, C and E are all estimated values:

1. When estimating the value of T_D ($\hat{T_D}$), D uses the estimated coordinates of B and C ($\hat{B}$ and $\hat{C}$). When the distance between B and C is smaller than $2 * 3\sigma$, the situation presented in Figure 7.16 a) can occur. For this case $\hat{T_D}$ will have a negative value. Because of this, for all the scenarios where $\hat{T_D} < 0$, the following equation is used which represents the minimum distance needed

between two estimated coordinates so that $\hat{T_D} > 0$:

$$\hat{T_D} = (2 * 3\sigma)/v. \tag{7.3}$$

Also, considering $\hat{T_D} < T_D$ the result is $\hat{T_{hello}} > T_{hello}$ which means that D joins the WSN later and this may reduce the total number of packets that D receives (if $T_{IC} < T_{RB}$). To make sure that this will not happen, $(2 * 3\sigma)/v$ will be added to $\hat{T_D}$ which means that it can be calculated with the following equation where $Dist(\hat{B,C})$ is the estimated distance between B and C:

$$\hat{T_D} = \frac{Dist(\hat{B,C})}{v} + \frac{(2 * 3\sigma)}{v}. \tag{7.4}$$

2. The second extreme scenario that can appear due to location errors is when calculating $\hat{T_{IC}}$ as it is presented in Figure 7.16 b) , where $\hat{E}$ and $\hat{C}$ are E and C estimates. This is when $\hat{T_{IC}} < 0$, so the following formula will be used where m is chosen to make sure to avoid the situation where D can send a hello packet after it passed M's communication range:

$$\hat{T_{IC}} = [(2 * 3\sigma)/v]/m \tag{7.5}$$

If $\hat{T_{IC}} > T_{IC}$ D can go through M's communication range without sending a hello packet, so $\hat{T_W}$ can be reduced to make sure that this case is avoided and it can be calculated with:

$$\hat{T_W} = (\hat{T_{IC}} - \frac{\hat{T_{IC}} + \hat{T_D}}{t_s + t_p} * t_p)/2 \tag{7.6}$$

For the next simulation Δ is varied between $(0m, 90m)$ and T_{hello} is considered to be $\hat{T_W}$ calculated with Eq. 7.6. The average of 18 different scenarios is considered where

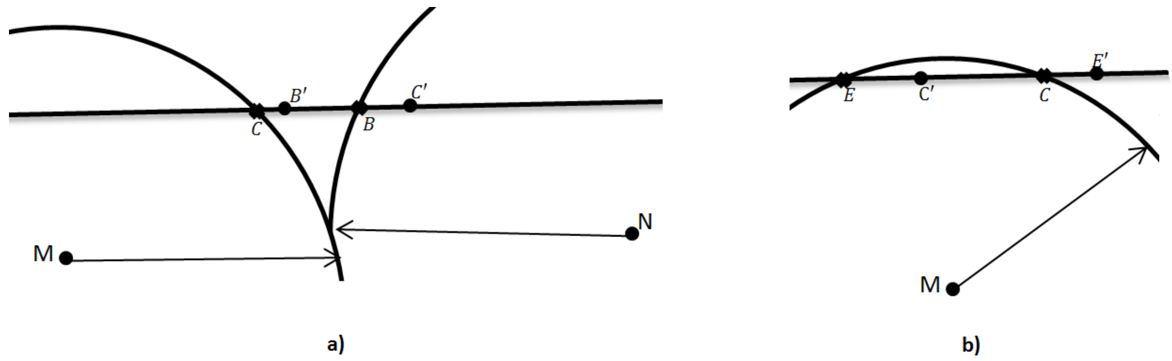

Figure 7.16: Extreme cases for calculating T_{hello}: a) Extreme case for T_D; b) Extreme case for T_{IC}

the D to WSN connectivity time varies from 10% to 95% using a 5% increment. For each scenario the average of 1000 simulations is made where ϵ follows a normal distribution. The obtained results are compared with the ones from PRP1 and presented in Figure 7.17. It can be seen that E is decreasing from +3.3% (for $\triangle = 0m$) to +0.4% (for $\triangle = 50m$ and $\triangle = 60m$) and then it rises to +0.7% (for $\triangle = 90m$). The T is rising from -3.7% (for $\triangle = 0m$) to -0.2% (for $\triangle \geqq 60m$). The results obtained for the T when $T_{hello} = \hat{T}_W$ and $\triangle = 60m$ in presence of location errors (which means that D has energy constraints) will be noted as PRP2.

The results obtained for PRP1, PRP2 and LAR3 are summarized in Table 7.5 and explained further:

- D needs to reduce its energy spent on hello packets when it has energy constraints. To do this it needs to broadcast hello messages only when it reaches its next neighbour communication area. Its current neighbour will approximate the time and place when D will rejoin the WSN. However, because of the location errors, D can start sending hello packets before entering its next neighbour communication range (for this case the number of hello packets which D sends to rejoin the WSN is high) or after it entered its neighbour communication range.

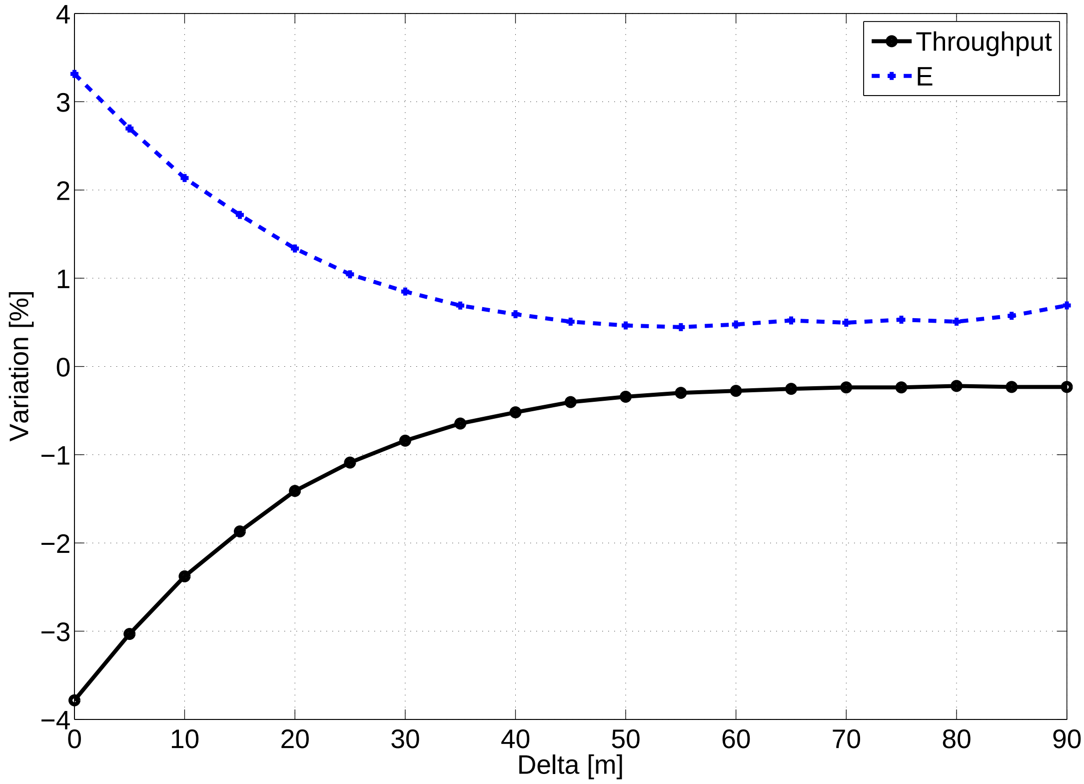

Figure 7.17: T and E variation when Δ is varied and $T_{hello} = T_{RS}$

- If $\epsilon < 0$ it means that D will start sending hello packets when it is already in its neighbour communication range. For this case, if $T_{IC} < T_{RB}$, D's neighbour will not have sufficient time to send all the packets from the buffer which means that, when compared with PRP1, the T will decrease. To compensate for the location error and to maximize the T, Δ is considered and based on ϵ variation is chosen to be equal to $3 * \sigma$

- If $\epsilon > 0$, D will start sending hello packets while it is outside its next neighbour communication range. If $\Delta = 3 * \sigma$ the time until D will join the WSN is maximized which means that more hello packets are sent by D and more energy is spent on this. The number of hello packets which are sent needs to be reduced to minimize the energy consumption. It was shown that, in order to obtain the best energy consumption, $T_{hello} = T_W$. PRP2 obtained a T decrease of 0.2% and 0.4% energy per packet increase when compared with PRP1.

For all the next simulations PRP2 and LAR3 are considered. PRP2 is the above discussed scenario in which D has energy constraints and will send hello packets when it is getting closer to its next neighbour communication range. LAR3 is the best case scenario for LAR when the time between sending two request packets was chosen to obtain the best trade off between E and T. These will be noted further as PRP and LAR for simplicity.

D con.	PRP1			PRP2			LAR3		
	T	AETED	E(J)	T	AETED	E(J)	T	AETED	E(J)
100%	19.2	0.052s	0.0006	19.2	0.052s	0.0006	19.2	0.052s	0.0006
10%-95%	17.48	0.057s	0.0006	17.43	0.057s	0.0006	10.03	0.099s	0.0052

Table 7.5: T, AETED and E for PRP1, PRP2 and LAR2 when D to WSN connectivity varies

7.3.3 The impact of WSN node density on PRP and LAR

The next simulations illustrate how E varies for both PRP and LAR when the WSN node density increases if the network size and D's speed remain constant. The average of 18 different scenarios is made where D to WSN connectivity time varies from 10% to 95% with a 5% increment. Also, for PRP, for each of the 18 scenarios the average of 1000 simulations is made where ϵ follows a normal distribution. The number of nodes is varied from 100 to 10,000 nodes using a 50 nodes increment. Increasing the number of nodes will only affect E and not the T because both PRP and LAR will use the shortest path from S to D to send data packets. The obtained results are presented in Figure 7.18. For the 100 nodes scenario PRP uses 88.5%

less energy to send a data packet from S to D and for the 10,000 nodes scenario 97.5% less energy. These results are summarized in Table 7.6.

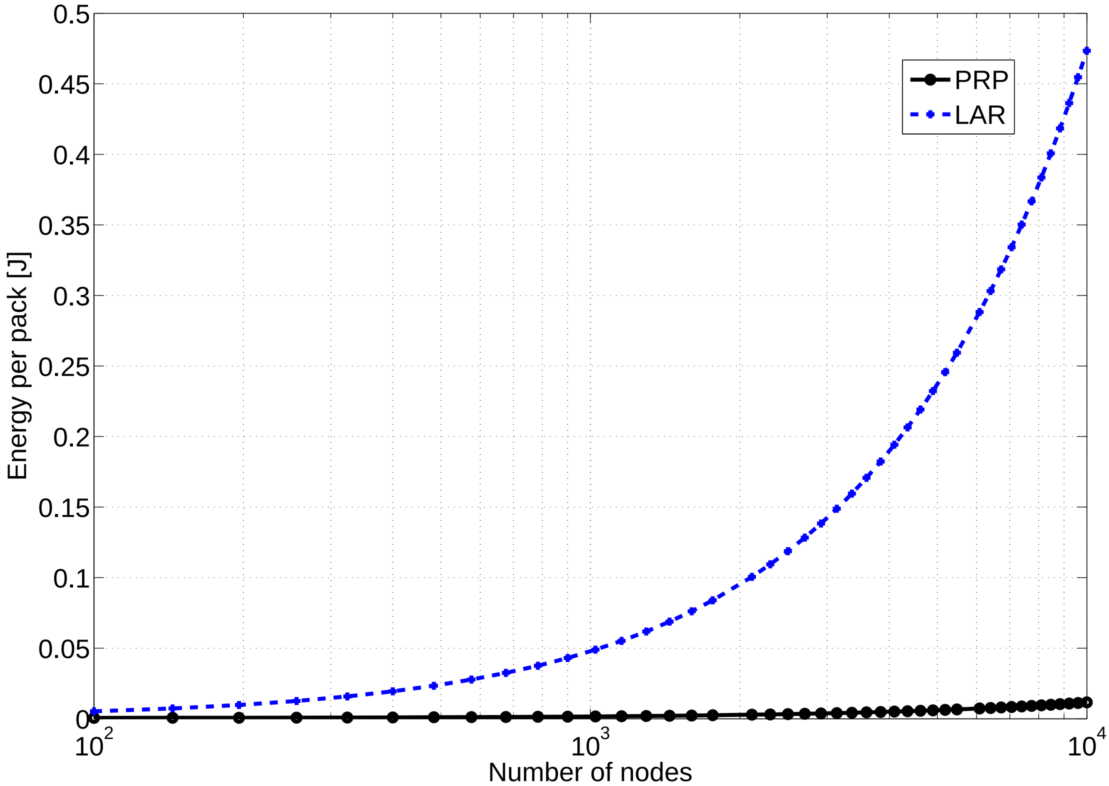

Figure 7.18: Nodes density impact on E for PRP and LAR

WSN density	PRP	LAR	PRP vs LAR
	E (J)	E (J)	Variation (%)
100 nodes	0.0006	0.0052	-88.5%
10000 nodes	0.0116	0.4733	-97.5%

Table 7.6: E when WSN density increases

These simulations prove that PRP is more suitable for high density networks when compared to LAR in terms of E. Both LAR and PRP performance are discussed next when the speed of D is varied.

7.3.4 The impact of D's speed on PRP and LAR

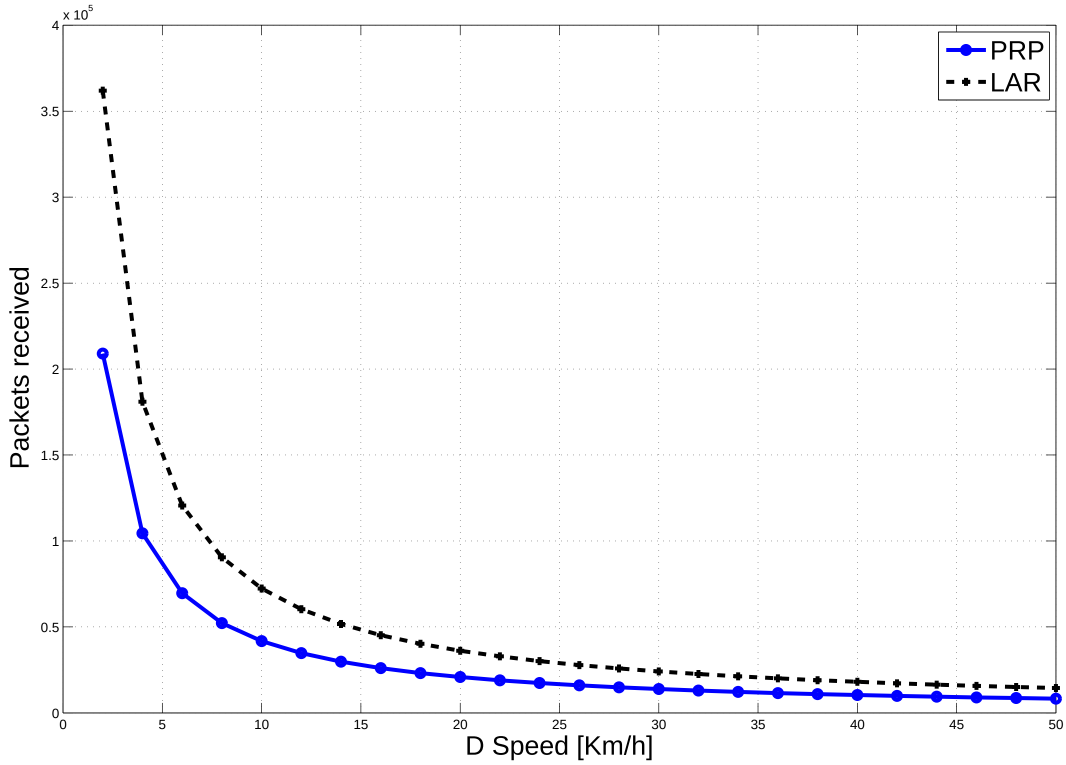

Figure 7.19: Received packets when D's speed varies

It will be analysed further how the T and E change for both PRP and LAR when D's speed increases from 2 km/h to 50 km/h using a 2 km/h increment. All the other network parameters are constant such as node density and WSN size. The average of 18 different simulations scenarios is considered where D to WSN connectivity varies from 10% to 95% with a 5% increment. For PRP, for each scenario, the average of 1000 simulations is made where ϵ follows a normal distribution. Each simulation time will allow for D to travel the entire WSN northern periphery from right to left. This means that time for each simulation is not the same and it will be influenced by D's speed (it will decrease when the speed increases). The total number of packets received at the destination for both PRP and LAR is presented in Figure 7.19 (not

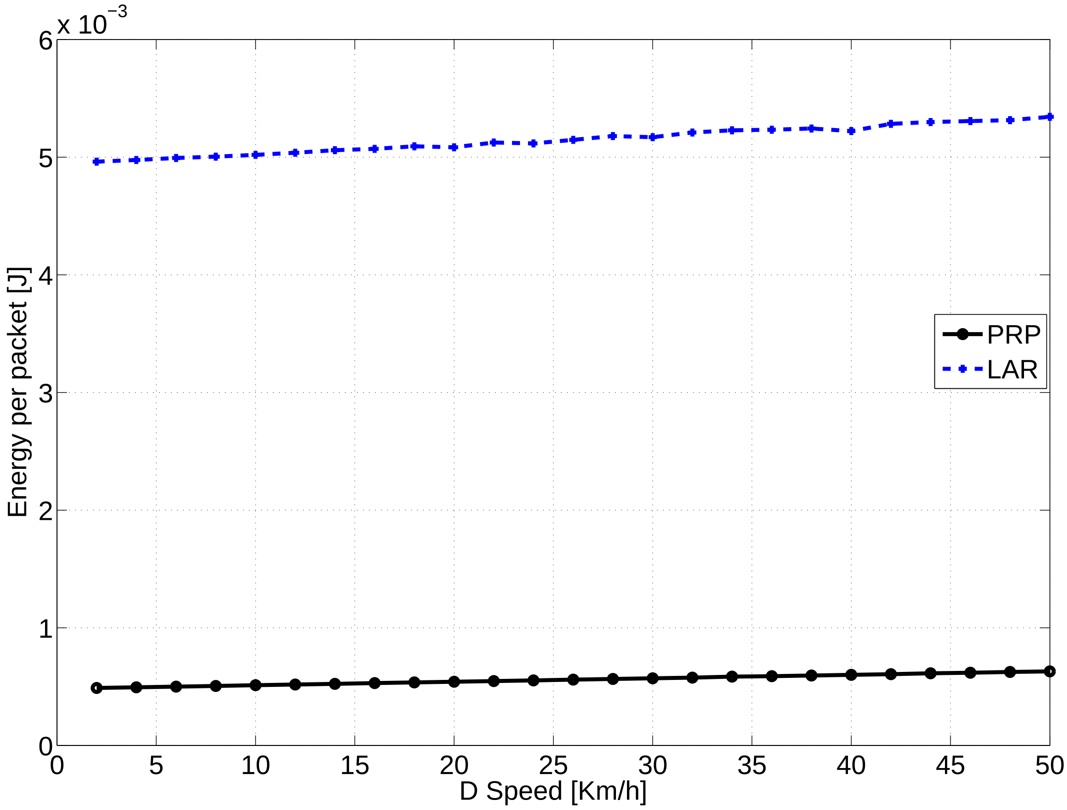

Figure 7.20: E variation when D's speed varies

the T because of its small variation not clearly visible on the graphic). The E is presented in Figure 7.20. The obtained results are summarized in Table 7.7. When D travels with 2 Km/h PRP sends with 73.2% more data packets using 90.3% less energy and for 50 Km/h it sends with 74.7% more data packets using with 88.4% less energy. The PRP improvements over LAR can be explained as follows:

- At high speeds, the distance that D travels connected with the WSN until a path to it is found increases. This distance will be bigger for LAR when compared with PRP. For LAR, S searches the path to D so until a route request packet reaches D and a route reply packet goes back to S, the distance that D travels in the WSN area until it establishes the connection with S increases when D's speed increases. For PRP, once D joins the WSN it will start to receive data packets directly from its neighbour buffer. So, the distance that D travels until it connects with the WSN is bigger for LAR when compared with PRP and it will affect the T and E.

The impact of the WSN size on the LAR and PRP performance is analysed in the next section.

Speed	PRP		LAR		PRP vs LAR	
	T	E(J)	T	E(J)	T	E
2Km/h	315.14	0.00048	181.94	0.00496	+73.2%	-90.3%
50Km/h	313.91	0.00062	179.69	0.00534	+74.7%	-88.4%

Table 7.7: PRP performance over LAR when D's speed varies

7.3.5 The impact of the WSN size on PRP and LAR.

The following setup is considered for the analysis of the LAR and PRP performance when the WSN size increases.

- The WSN size will be varied between 800m and 8000m in length and width with an 100m increment, so 44 simulations are made. It is important to mention that the simulation time will be constant (D's speed is constant at 40 km/h) which means that D will move at the WSN periphery for a fixed period of time of 57.4 seconds for each of the above 44 simulations. For each of these simulations the D to WSN connectivity varies between 10% and 95% with a 5% increment, so each simulation is an average of 18 different connectivity scenarios. Also, for PRP, for each scenario, the average of 1000 simulations is made where ϵ follows a normal distribution.
- The number of nodes from the WSN will increase with the WSN size, which means that, if for a 800 m^2 WSN there are 100 nodes, then for a 1600 m^2 there will be 400 nodes.

Two different WSN scenarios are considered further as follows:

- In the first scenario (noted S1) S sends 20 packets per second. This means that the route path increase will affect only the time after the first packet is received by D each time D rejoins the WSN, all the other packets arriving at the destination after 0.05 seconds.
- In the second scenario (noted S2) S sends a data packet only after it receives a confirmation from D that the previous sent packet was received. This means that the route path increase between S and D will affect the time after each packet is received by D.

In Figure 7.21 and Figure 7.22 are presented the T and E variation when the WSN size is varied for both S1 and S2. Because the simulation time is kept constant, less packets will arrive at the destination when the path between S and D increases (when the WSN increases). However, it can be seen that for S2 the T drop is much faster for both PRP and LAR because less packets will arrive at the destination if only one packet is on the fly between S and D. In Table 7.8 and Table 7.9 the T, E and AETED is summarized for PRP and LAR (S1 and S2) for two WSN sizes of $800m^2$ and $800m^2$.

Size	PRP2			LAR			PRP2 vs. LAR		
	T	AETED	E(J)	T	AETED	E(J)	T	AETED	E
$800m^2$	17.44	0.057s	0.0006	10.03	0.099s	0.0052	73.8%	-42%	-88.5%
$8000m^2$	17.1	0.058s	0.0166	8.57	0.116s	0.0871	99.6%	-50%	-81%

Table 7.8: Scenario1: T, AETED and E when WSN size increases

Size	PRP2			LAR			PRP2 vs. LAR		
	T	AETED	E(J)	T	AETED	E(J)	T	AETED	E
$800m^2$	93.05	0.011s	0.0005	55.06	0.019s	0.0014	69%	-41%	-63%
$8000m^2$	8.83	0.114s	0.0274	4.36	0.229s	0.165	103%	-50%	-84%

Table 7.9: Scenario2: T , AETED and E when WSN size increases

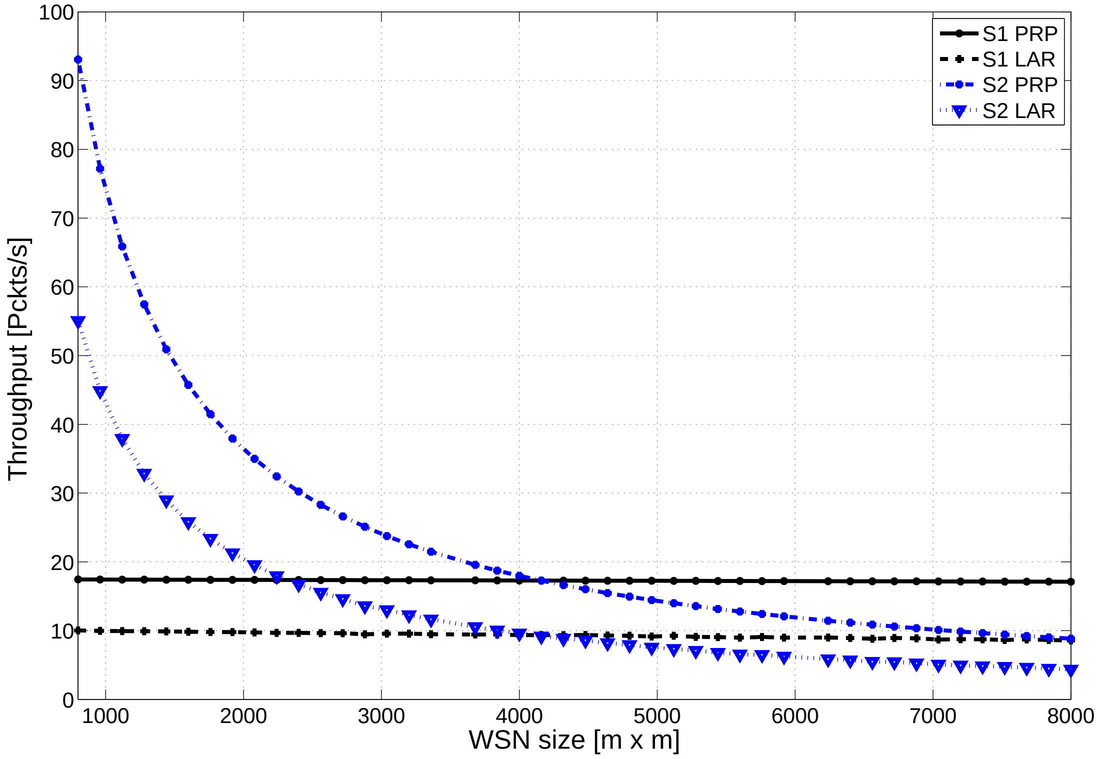

Figure 7.21: T for LAR and PRP (S1 and S2)

The following observations can be made:

- For the 800 m^2 scenario PRP sends 73.8% more data packets for S1 and 69.7% more for S2. For S1 it uses 88.5% less E and for S2 64%. This difference appears because for a 800 m^2 S2 scenario the path between S and D has on average 12 hops which means that S sends 102 data packets per second. When the D to WSN connectivity time varies, for scenarios bellow 50% connectivity there is not sufficient time for D to receive all the packets sent to its neighbour buffer. So, more packets are sent to D's neighbour buffer when compared with the packets that D can receive from it when it will re-join the WSN.

- For large networks, the 8000 m^2 scenario, for S2, the path from S to D increases to an average of 120 hops. D will have time to receive all the buffered packets

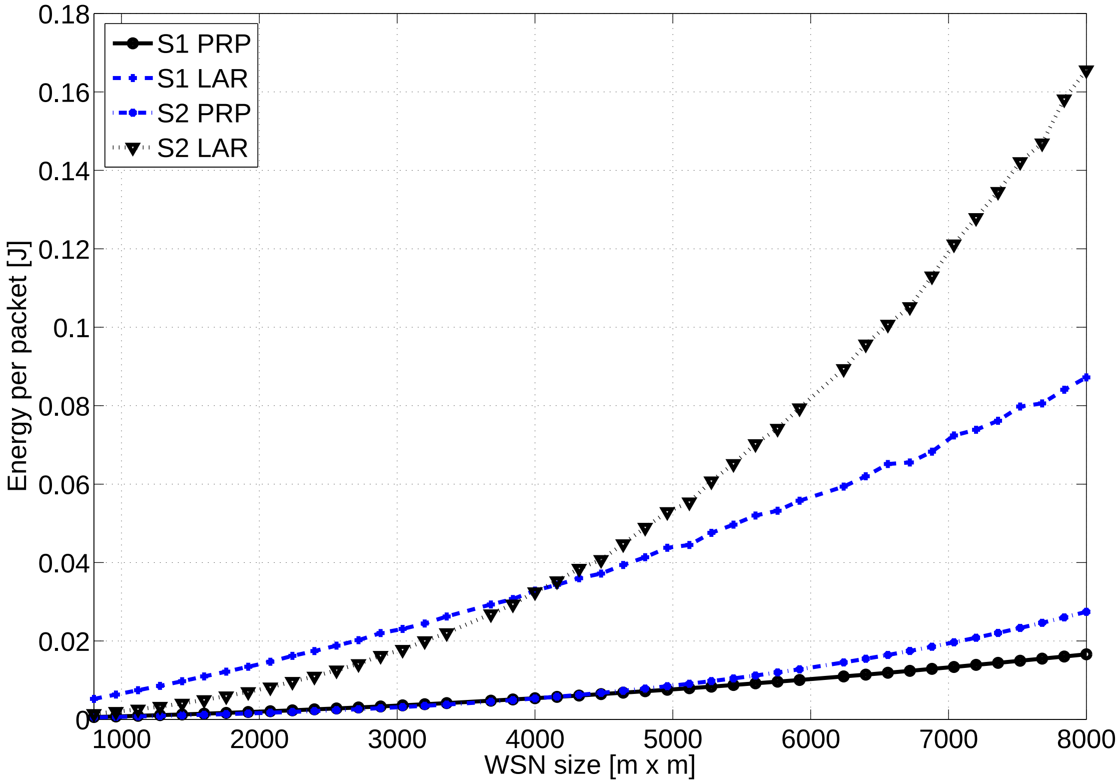

Figure 7.22: E for LAR and PRP (S1 and S2)

from its neighbour for all the 10% to 95% connectivity scenarios which means that when compared to S1, the PRP vs LAR improvements are almost the same.

7.4 Conclusion

This chapter presented the MATLAB simulator used to compare the performance of LAR, AODV, DSR and PRP. LAR and PRP routing diagrams and pseudo-code are described in section 7.1.1 and Appendix 1.6 and 1.7. The obtained results were analysed and discussed for variations of several network parameters: the connectivity of D to the WSN, D speed, WSN size and WSN node density. The obtained results

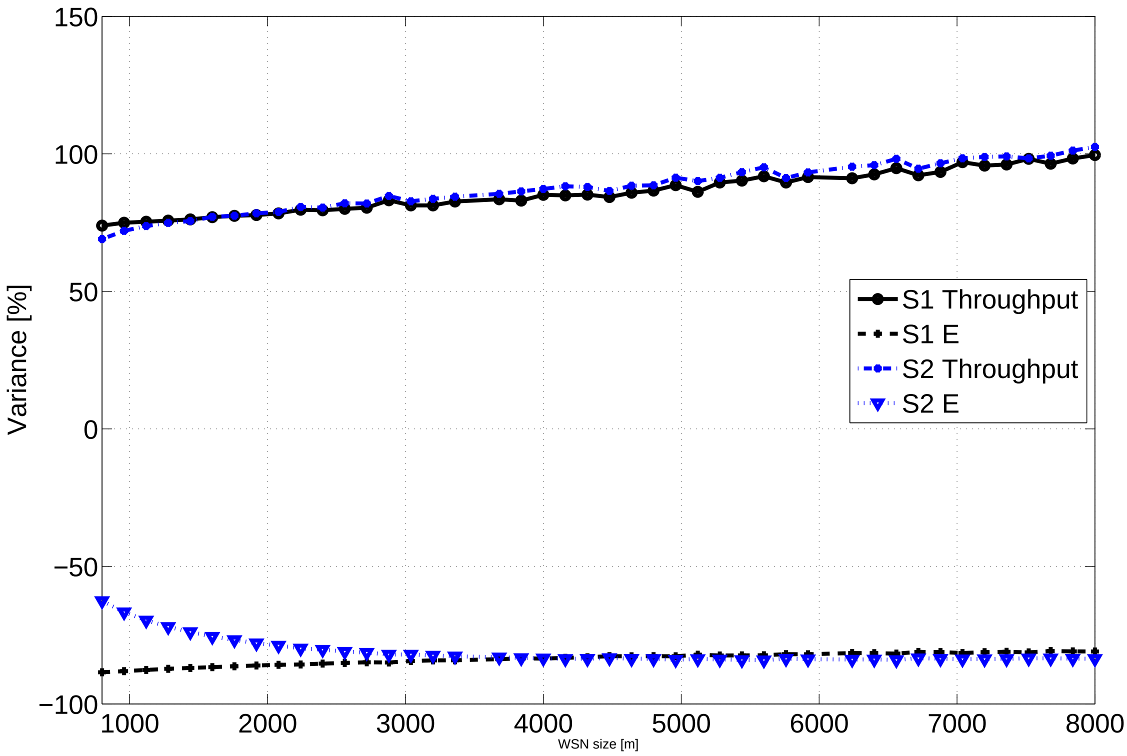

Figure 7.23: PRP over LAR improvements (E and T) for S1 and S2

can be summarized as follows:

- When D has no energy constraints:
  - PRP has a 73% T improvement over LAR if $t_s > 0.04$ seconds. This means that D has time to receive all its neighbour buffered data packets.
  - Overall, across the 10% - 95% connectivity scenario, PRP obtains an average T of 17.48, LAR and AODV of 10.1 and DSR of 10.05. LAR and AODV have an AETED increase of 88.4%, DSR of 89.6% and PRP of 9.6% when compared with the 100% connectivity scenario. Also PRP maintains the same E but AODV and DSR have an increase of 33% and LAR of 30%. PRP has the best results because once D rejoins the WSN it will start to receive packets from its neighbour buffer. This is pos-

sible because the peripheral routing algorithm that PRP uses estimates through which node D will rejoin the WSN. Based on these estimates S will send packets to D's future neighbour until D rejoins the WSN.

  - Increasing T_{RREQ} will reduce the E for LAR, but also its T hence a compromise needs to be considered when choosing this value. $T_{RREQ} = 12 * t_0$ is considered to be a good compromise for which LAR uses 90.33% less E and has a decrease in T of 0.8%.

- When D has energy constraints:
  - To save power D needs to broadcast hello packets when it is getting closer to its neighbour coverage area. Also T_{hello} needs to be carefully chosen to minimise the number of hello packets that D sends. Using the equations presented in section 8.3.1 and 8.3.2 PRP has a 0.4% E and 0.2% T increase when compared with the scenario in which D has no energy constraints.
  - PRP is more suitable for high density networks when compared with LAR because it uses with 97.5% less E for a 10.000 nodes scenario. This is possible because LAR sends RREQs in the WSN to search a path to D until D joins the WSN and PRP sends just one RREQ to search a path to D's next neighbour.
  - When D's speed increases to 50 Km/h, PRP has a T improvement over LAR of 74.7% and it uses with 88.4% less E. At high speeds the distance that D travels connected with the WSN until a path to it is found increases for LAR. For PRP this will remain approximately the same because D will start to receive data packets from its neighbour buffer.
  - For the scenario in which one data packet is on the fly between S and D,

when the WSN size increases to 8000m in length and width, PRP has a 103% T improvement and is using with 84% less E. For the same size when more data packets are on the fly between S and D, PRP has a 99.6% T improvement and is using with 81% less E.

The next chapter presents the conclusions of this thesis and possible future work.

8 Conclusions and future work

In this chapter the conclusions are presented together with thesis'As novelty, applicability, future work and constraints.

8.1 Conclusions

The thesis starts with a short WSNs introduction made in chapter 1. WSNs applications are discussed in section 1.2 with an emphasis regarding military applications, building monitoring, habitat monitoring and industrial applications. WSN specifications and constraints are presented in section 1.3 where factors which influence the network design are detailed, like: fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media and power consumption. The thesis motivation, novelty and contribution together with thesis outline and publications are presented further in the next sections.

Chapter two starts with a discussion about the need of mobile sinks in WSNs. Different mobility models are presented and compared in section 2.1. The MMM is introduced and its main characterisctics are discussed in section 2.2.

In chapter three a comprehensive discussion is made about WSN sinks mobility support. Routing protocols classifications and specifications are given is section 3.1. Different routing protocols and location prediction schemes proposed to support mobile sinks are presented in section 3.2 and section 3.3 analysing if the MMM characterisctis are supported. At the end of this chapter (section 3.4) a discussion is made about the MMM and the delay tolerant characteristics that a WSN has when a sink follows this mobility model.

In chapter four it is shown through Qualnet simulations that the total number of sent packets from a source node to a mobile sink decreases considerably when the distance between the sink and WSN increases (the mobile sink follows the MMM). Five MANET routing protocols are chosen to send the data: LAR, AODV, DSR, Bellman Ford and DYMO. How these protocols work is explained in section 4.2. The simulation topology used for the Qualnet simulator is presented in section 4.3 where a military application is considered as an example in which a mobile sink needs to collect information from the battlefield area (WSN area) without entering the war zone. For this, the mobile sink will move at the WSN periphery so, during its movement it will join and leave the WSN frequently. The simulation results from this chapter can be summarized as follows: LAR sends the highest number of packets when the distance between the sink and WSN is maximum however the average end to end delay is also high in this case; for high sink speeds LAR perform better but again the high values obtained for the average end to end delay concludes that none of these five protocols support the MMM. MAC and routing layer Interactions are discussed and analysed for the proposed application scenario in section 4.3.1 which leads to the conclusion that different MAC protocols have different results so this will influence the total number of packets sent to the sink. This chapter ends with the proposition that a new routing protocol needs to be designed in order to consider the specific characteristics that a mobile sink has when it follows the MMM.

The “Peripheral Routing Protocol” meant to support the MMM is introduced in chapter five. A comprehensive explanation about how it works is given in the beginning. Details about when PRP needs to be used follow in section 5.1. How PRP estimates next neighbour ID and how the destination's deconnectivity time from the WSN is estimated is discussed and analysed in section 5.2 and section 5.3. This chapter ends with a presentation about PRP recovery measures meant for extreme conditions.

In chapter six both LAR and PRP analytical models for the throughput and energy are explained and compared. Throughout mathematical analysis the following conclusions are stated: for the scenario in which D travels disconnected from the WSN and more then two hops are between S and D, PRP will send more data packets when compared with LAR; the energy spent on RREQ by LAR is bigger than the energy spent on hello packets plus the energy to find a path from S to D by PRP. These are shown through MATLAB simulations in the following chapter.

In chapter seven the MATLAB simulation results are presented and discussed for a military application in which a mobile sink that needs to receive information from a source inside a WSN moves at the periphery following the MMM. The chapter starts with a presentation of the used MATLAB simulator and both LAR and PRP routing diagrams. The simulation scenario is presented next and two situations are considered: when D has and has not energy constraints (for energy constraints the location errors are taken into consideration). LAR, AODV, DSR and PRP are considered for the first scenario (D has no energy constraints) where D to WSN connectivity time varies between 10% and 95%. PRP outperforms all the routing protocols because it sends more data packets to D faster and when compared with LAR the following results are obtained: it sends with 73% more data packets, 41.8% faster using 98.8% less energy. Further, in section 7.3 a discussion is made on how to choose the optimal values for Δ and T_{hello} to have the highest T and lowest E for the case when D has energy constraints. When compared with the first case (where D

does not have energy constraints) the following results are obtained for PRP: a 0.2% T decrease and a 0.4% E increase. Considering PRP and LAR, different simulations are made further where: the WSN node density increases from 100 to 10,000 nodes (PRP uses with 97.5% less energy to send a packet from S to D); D speed increases from 2 km/h to 50 km/h (PRP sends 74.7% more packets using 88.4% less energy); the WSN size increases from 800m to 8000m in length and width and two scenario are considered: a first scenario in which more data packets are on the fly between S and D; a second scenario in which one packet is on the fly between S and D. For the first scenario, when compared with LAR, PRP sends 99.6% more data packets using 81% less energy and for the second scenario it sends 103% more data packets using 84% less energy.

So, it can be concluded that PRP outperforms LAR for a scenario in which D follows the MMM and for which the following parameters are varied: D to WSN connectivity and speed, WSN size and density. Thesis novelty is presented in the next section.

8.2 Novelty

A military application scenario was considered where senors are deployed over a battlefield area. The mobile sink needs to collect information from the formed WSN without entering the battlefield zone. Because of this, it will need to move at the network periphery and during its movement it will join and leave the WSN area frequently. Research was made in the WSN routing protocols area and a survey was published based on this research in [36]. WSN mobility models were studied and a new mobility model (MMM) where a sink moves at the network periphery was defined. It was shown through Qualnet simulations that five MANETs routing protocols fail to send data packets to a sink which follows the MMM. This work was published in [35]. A new routing protocol (PRP) was designed to support the MMM and through MATLAB simulations it was shown that it outperforms protocols like LAR, AODV and DSR in terms of T, AETED and E. Analytical models were defined for T and E for both LAR and PRP and based on a mathematical analysis it is stated that PRP outperforms LAR. Also, more simulations were made to compare the performances of these two protocols where different parameters are varied like: D to WSN connectivity time, D speed, WSN size and density. This work was submitted to the Elsevier journal. The PRP applicability is discussed further.

8.3 Applicability

The PRP was designed for applications in which a mobile sink needs to collect information from an area inaccessible to humans or machines. The area can be inaccessible because of several factors like: pollution, radiation, rough terrain, restricted access (habitat monitoring) or any other factors like fire, flooding, etc.. Also, using a mobile unit at the WSN periphery will ensure path diversity which improves the overall WSN lifetime. It was discussed that a mobile node which moves at the WSN periphery will connect and disconnect with the WSN are frequently following the MMM characteristics (described in sub-chapter 2.2). So, the PRP was designed to support these characteristics. Based on the PRP description from chapter 5, PRP mathematical analysis from chapter 6 and PRP routing diagrams together with the pseudo-code from chapter 7, one can implement this protocol for a real application scenario. Also the analysis and results presented in this thesis can help identify the main challenges that PRP will face for different application characteristics.

Scenario applications in which PRP can be used are given as follows:

- Military applications - the collected data from the battlefield needs to reach the destination as fast as possible. This data can be vital in saving lives, taking military strategic actions or making accurate evaluations of a current situation. So, the collected data needs to reach the destination fast. In order to minimise the material or life losses a mobile unit (machine or human) can be used at the battlefield periphery (WSN periphery) to receive the data. For this case, using PRP will assure a good T, and low values for both AETED and E.

- Habitat monitoring - sensors can be spread over a wide area to track animals behaviour. This behaviour can be affected by human or machine presence so it is recommended to collect data outside the monitored environment. This will be possible by having a mobile unit at the WSN periphery. PRP can support this type of movement and can assure a high T with minimal E.

- Building Monitoring - sensor nodes can be used for heat or smoke monitoring in a building to detect or prevent fires. However in case of a fire the building can become inaccessible to humans or machines. Receiving constant updates from the building about the situation is vital for firemen to be able to make a strategy on how to reduce to a minimum the human and material losses. For this scenario, the firemen are situated outside the building and throughout their movement can enter and leave the building's WSN coverage area frequently. For this case the PRP can send fast the data to the mobile units assuring a high T and a low E.

Future work and constraints are presented in the next section.

8.4 Future work and constraints

Future work can be carried out on the following aspects:

- The PRP was analysed for the scenario where D moves in a straight line at a constant speed for each set of simulations. Different scenarios need to be considered when both D's speed and direction vary.
- At low D's speeds, certain paths can be used in excess which would lead to energy depletion for certain nodes. Because of this, an energy management mechanism needs to be implemented where nodes can request to be removed from a path and replaced by one of their neighbours if the energy level reaches a certain minimum threshold.
- The nodes distribution used was a normal one for all the simulations. However, different distributions need to be considered and the WSN lifetime analysed. A distribution in which nodes are grouped around different paths that are formed between S and D when D follows the MMM is expected to improve the overall WSN life time.
- The scenario with multiple sources and multiple destinations can be considered also for future work.
- The MMM was analysed only for the case where the movement is made at the WSN periphery. However, the MMM model can also be met for a scenario in which the number of coverage holes in the WSN is high and D moves throughout the network. This case can be analysed in future research.
- It is known that MAC and routing layer interactions affect the overall performance. So, simulations need to be made for both PRP and MAC to see how the performance changes.

- Future work can be carried out to implement and test PRP performances for a real case scenario.

PRP constraints are listed bellow:

- It was shown that both PRP and LAR have the same performance for the scenario where D moves all the time connected with the WSN. The current PRP implementation does not take into account the situation were D can change its direction of movement frequently. For the 100% connectivity scenario, where D changes the movement direction, it is expected to have better performances for LAR over PRP. A threshold between D to WSN connectivity and D frequency of changing the direction of movement needs to be established in order to obtain the best results in term of T, AETED and E. Also, a situation where both protocols are used is a good option: LAR can be used for high D to WSN connectivity values and PRP for the low ones.

9 Appendices

9.1 Appendix 1 - calculating $\mathbf{Pckts_{LAR}}$ when $\mathbf{T_D = 0}$ and $\mathbf{T_{wait} = 0}$

$$Pckts_{LAR} = 1 + \left[\frac{T_{RC} - t_p * n_p}{t_s}\right]$$

$T_{RC} = T_{IC} - \tau_L - T_{RREP}$ and $\tau_L = \left(\left[\frac{T_D}{\xi_L}\right] + 1\right) * \xi_L - T_D$. Also $\xi_L = T_{RREQ} + T_{wait}$ and if $T_D = 0$ and $T_{wait} = 0$ then $\tau_L = T_{RREQ}$. By replacing τ_L in $Pckts_{LAR}$ we obtain:

$$Pckts_{LAR} = 1+\left[\frac{T_{IC} - \tau_L - T_{RREP} - t_p * n_p}{t_s}\right] \Leftrightarrow Pckts_{LAR} = 1+\left[\frac{T_{IC} - T_{RREQ} - T_{RREP} - t_p * n_p}{t_s}\right]$$

9.2 Appendix 2 - calculating $\tau_\mathbf{L}$

$$\tau_L = \left(\left[\frac{T_D}{\xi_L}\right] + 1\right) * \xi_L - T_D = \left(\left[\frac{T_D}{T_{RREQ} + T_{wait}}\right] + 1\right) * (T_{RREQ} + T_{wait}) - T_D$$

If it is assumed that $T_{wait} = 0$ then:

$$\tau_L = \left(\left[\frac{T_D}{T_{RREQ}}\right] + 1\right) * T_{RREQ} - T_D$$

$T_{RREQ} = t_p * n_{max}$ where t_p is the time to send a packet between two nodes and n_{max} is the length from S to the further node from the WSN. τ_L becomes:

$$\tau_L = \left(\left[\frac{T_D}{t_p * n_{max}}\right] + 1\right) * t_p * n_{max} - T_D$$

If it is assumed that T_D is multiple of T_{RREQ}, $T_D = m*T_{RREQ}+\psi$ where $\psi < t_p*n_{max}$ then:

$$\tau_L = \left(\left[\frac{m * t_p * n_{max} + \psi}{t_p * n_{max}}\right] + 1\right) * t_p * n_{max} - m * t_p * n_{max} \Leftrightarrow$$

$$\tau_L = m * t_p * n_{max} + t_p * n_{max} - m * t_p * n_{max} \Leftrightarrow$$

$$\tau_L = t_p * n_{max}$$

9.3 Appendix 3 - solving $\mathrm{Pckts_{PRP}} - \mathrm{Pckts_{LAR}} = 0$

$$1+\left[\frac{T_D + \tau_P - t_p * (n_p - 1)}{t_s}\right]+\left[\frac{T_{IC} - (\tau_P + T_{ec} + T_{RB})}{t_s}\right]-1-\left[\frac{T_{IC} - (\tau_L + T_{RREP} + t_p * n_p)}{t_s}\right] = 0$$

$$\Leftrightarrow \frac{T_D + \tau_P - t_p * (n_p - 1) + T_{IC} - (\tau_P + T_{ec} + T_{RB}) - [T_{IC} - (\tau_L + T_{RREP} + t_p * n_p)]}{t_s} = 0$$

$$T_D + \tau_P - t_p * n_p + t_p + T_{IC} - \tau_P - T_{ec} - T_{RB} - T_{IC} + \tau_L + T_{RREP} + t_p * n_p = 0$$

$$T_D + t_p - T_{ec} - T_{RB} + \tau_L + T_{RREP} = 0$$

$$T_D + t_p + \tau_L + T_{RREP} = T_{ec} + T_{RB}$$

However $T_{RB} = N_b * t_p \Leftrightarrow T_{RB} = \{1 + [{}^{T_D + \tau_P - t_p * (n_p - 1)}/_{t_s}]\} * t_p$ so we obtain:

$$T_D + t_p + \tau_L + T_{RREP} = T_{ec} + t_p + \frac{T_D + \tau_P - t_p * (n_p - 1)}{t_s} * t_p$$

$$T_D * \left(1 - \frac{t_p}{t_s}\right) = T_{ec} + \frac{t_p * [\tau_P - t_p * (n_p - 1)]}{t_s} - t_p - \tau_L - T_{RREP}$$

When a path is found $T_{RREQ} = T_{RREP}$ $(n_{max} = n_p)$. If $T_{wait} = 0$ then $\tau_L = T_{RREQ} = t_p * n_p$ (τ_L is calculated in Appendix 2) and $\tau_P = T_{hello} = t_p$. $T_{ec} = 2 * t_p$. If ${}^{t_p}/_{t_s} = {}^1/_x$ $(t_s = x * t_p)$ then $Pckts_{PRP} - Pckts_{LAR}$ becomes:

$$Pckts_{PRP} - Pckts_{LAR} = T_D * \left(1 - \frac{1}{x}\right) - \left\{T_{ec} + \frac{[\tau_P - t_p * (n_p - 1)]}{x} - t_p - \tau_L - T_{RREP}\right\}$$

$$\Leftrightarrow Pckts_{PRP} - Pckts_{LAR} = T_D * \left(1 - \frac{1}{x}\right) - 2 * t_p - \frac{1}{x} * (2 * t_p - t_p * n_p) + t_p + 2 * t_p * n_p$$

$$\Leftrightarrow Pckts_{PRP} - Pckts_{LAR} = T_D * \left(1 - \frac{1}{x}\right) - t_p - \frac{2 * t_p}{x} + \frac{t_p * n_p}{x} + 2 * t_p * n_p$$

$$\Leftrightarrow Pckts_{PRP} - Pckts_{LAR} = T_D * \left(1 - \frac{1}{x}\right) + \frac{t_p}{x} * (n_p - 2) + t_p * (2 * n_p - 1)$$

However, it can be seen that $Pckts_{PRP} - Pckts_{LAR} > 0$ if: $T_D > 0$, and $n_p > 1$

9.4 Appendix 4 - solving $\mathbf{E_{LAR_{RREQ}} = E_{PRP_{hello}} + e * (m * n_r + m * n_o + n_o - 1)}$

$$E_{LAR_{RREQ}} = \left(\left[\frac{T_D}{T_{RREQ} + T_{wait}}\right] + 1\right) * e_{rreq} * n_n$$

If $T_D = m * n_p * t_p + \psi$, $\psi < n_p * t_p$, $T_{RREQ} = t_p * n_p$ and $T_{wait} = 0$ then:

$$E_{LAR_{RREQ}} = \left(\left[\frac{m * n_p * t_p + \psi}{t_p * n_p} \right] + 1 \right) * e_{rreq} * n_n$$

$$\Leftrightarrow E_{LAR_{RREQ}} = n_n * e_{rreq} * (m + 1)$$

But $n_n = n_m + n_o$ and $e_{rreq} = e$ so $E_{LAR_{RREQ}}$ becomes:

$$\Leftrightarrow E_{LAR_{RREQ}} = (n_m + n_o) * e_{rreq} * (m + 1)$$

$$\Leftrightarrow E_{LAR_{RREQ}} = (n_m * e + n_o * e) * (m + 1)$$

$$\Leftrightarrow E_{LAR_{RREQ}} = m * n_m * e + m * n_o * e + n_m * e + n_o * e$$

If we consider that $n_m = n_p + n_r$ then $E_{LAR_{RREQ}}$ becomes:

$$E_{LAR_{RREQ}} = m * (n_p + n_r) * e + m * n_o * e + (n_p + n_r) * e + n_o * e$$

$$\Leftrightarrow E_{LAR_{RREQ}} = m * n_p * e + m * n_r * e + m * n_o * e + n_r * e + n_p * e + n_o * e$$

The equation for $E_{PRP_{hello}} + E_{S_{RREQ}}$ is:

$$E_{PRP_{hello}} + E_{S_{RREQ}} = \left(\left\lceil \frac{T_D}{T_{hello} + T_{wait}} \right\rceil + 1\right) * e_{hello} + n_m * e_{rreq}$$

Also $T_{hello} = t_p$ and $e_{hello} = e$ so:

$$E_{PRP_{hello}} + E_{S_{RREQ}} = \left(\left\lceil \frac{m * n_p * t_p + \psi}{t_p} \right\rceil + 1\right) * e + n_m * e$$

$$\Leftrightarrow E_{PRP_{hello}} + E_{S_{RREQ}} = (m * n_p + 1) * e + n_m * e$$

$$\Leftrightarrow E_{PRP_{hello}} + E_{S_{RREQ}} = m * n_p * e + e + n_m * e$$

However, $n_m = n_p + n_r$ so we can write that:

$$E_{PRP_{hello}} + E_{S_{RREQ}} = m * n_p * e + e + n_p * e + n_r * e$$

So, we can conclude that:

$$E_{LAR_{RREQ}} = E_{PRP_{hello}} + E_{S_{RREQ}} + e * (m * n_r + m * n_o + n_o - 1)$$

9.5 Appendix 5 - solving $\mathbf{Pckts_{PRP}} = \mathbf{Pckts_{PRP_E}}$

$Pckts_{PRP} = Pckts_{PRP_E}$ can be written as follows:

$$\left\lceil \frac{T_{IC} - \tau_p - T_{EC}}{t_p} \right\rceil = \left\lceil \frac{T_{IC} - \tau_p - T_{EC} - T_{error}}{t_p} \right\rceil$$

However $T_{IC} - \tau_p - T_{EC} = n * t_p + \psi$, $\psi < t_p$ which means that :

$$\left\lceil \frac{n * t_p + \psi}{t_p} \right\rceil = \left\lceil \frac{n * t_p + \psi - T_{error}}{t_p} \right\rceil$$

$$\Leftrightarrow \psi - T_{error} < t_p \Leftrightarrow T_{error} < \psi < t_p$$

9.6 LAR pseudocode

LAR pseudocode is presented bellow together with its main functions:

1. BEGIN;

2. WSN_setup;

```
WHILE (No_packets > 0)
  IF (path_found = false) & (D_coordinates = false)
    RREQ_area = WSN_area;
    REPEAT
      [path_found, path] = send_RREQ(S, D, RREQ_area);
      wait(T_rreq);
      No_RREQs = No_RREQs + 1;
    UNTIL (path_found = true) || (No_RREQ = limit)
  END IF
  IF (path_found = false) & (D_coordinates = true)
    RREQ_area = area(S,D);
    REPEAT
      [path_found, path] = send_RREQ(S, D, RREQ_area);
      wait(T_rreq);
      No_RREQs = No_RREQs + 1;
    UNTIL (path_found = true) || (No_RREQ = limit)
  END IF
  IF (No_RREQs = limit)
    No_RREQs=0;
    wait(T_retry);
    RREQ_area = WSN_area;
  END IF
```

```
25.     IF (path_found = true)
26.        No_RREQs = 0;
27.        REPEAT
28.           [path_found] = send_data(S, D, path);
29.           No_packets=No_packets-1;
30.           wait(Ts);
31.        UNTIL (No_packets = 0) || (path_found = false)
32.     END IF
33.  END WHILE
34.  END
```

The following explanation is given about the functions and parameters used in the above pseudocode program:

1. Start the program.

2. “WSN_setup” is a function in which the following WSN parameters are defined: network area width and length, number of nodes, number of packets, *D*’s speed, node transmission range, node transmit power, path loss exponent, the standard deviation for shadowing model, sensitivity threshold, route request packet size, route reply packet size, hello packet size, data packet size, data rate, energy per bit spent on transmission, energy per bit spent for reception, initial node energy.

3. The while syntax from 3 to 33 will search a path from S to D and if a path is found will send data packets as long as “No_packets $>$ 0” (where No_packets represent the total number of packets).

4. Check if a path from S to D is not known nor D previous coordinates when D was connected with the WSN.

5. The RREQ area in which S sends RREQ packets becomes the entire WSN.

6. A path from S to D is searched.

7. A RREQ packet is sent from S to D in the RREQ_area. When a path is found "path_found" is returned true and the found path in "path".

8. The waiting time between two RREQs packets is "T_rreq" seconds.

9. The number of RREQs sent in the WSN are counted by "No_RREQ".

10. RREQs are sent until a path is found ("path_found = true") or a certain number of RREQs are sent (No_RREQ = limit).

11. It ends the case for which a path from S to D is searched when D coordinates are not known.

12. Checks if a path from S to D is not known but D previous coordinates when D was connected with the WSN are known.

13. The RREQ area in which S sends RREQ packets is the one determined by S and D coordinates.

14. A path from S to D is searched.

15. A RREQ packet is sent from S to D in the RREQ_area. When a path is found "path_found" is returned true and the path is returned in "path".

16. The waiting time between two RREQs packets is "T_rreq" seconds.

17. The number of RREQs sent in the WSN are counted by "No_RREQ".

18. RREQs are sent until a path is found ("path_found = true") or a certain number of RREQs are sent (No_RREQ = limit).

19. It ends the case for which a path from S to D is searched when D previous coordinates when it was connected with the WSN are known.

20. Checks if the maximum number of RREQs packets have been sent in the WSN.

21. No_RREQs becomes 0.

22. If the maximum number of RREQs packets have been sent in the network, S waits for a certain period of time (T_retry seconds) after which it will restart sending RREQs.

23. The RREQ area becomes the entire WSN.

24. It ends the case when No_RREQs has reached the maximum value.

25. Checks if a path from S to D is found.

26. The number of RREQs sent in the WSN is set to 0.

27. S starts to send data packets to D.

28. A data packet is sent from S to D. "path_found" becomes false when D leaves the WSN area (in this case D neighbour informs S to stop data transmission because the path is broken) or the path is broken by different reasons (other nodes from the path are not available).

29. "No_packets" represents the total number of packets which needs to be sent.

30. Between two data packets transmission S waits for Ts seconds.

31. S sends data packets to D until all the packets are sent (No_packets = 0) or when the path between S and D breaks (path_found = false).

32. The scenario for which a path is known between S and D (or D next neighbour) and data packets are sent ends.

33. The scenario for which S has packets to send ends.

34. End program.

9.7 PRP pseudocode

PRP pseudocode is presented bellow together with its main functions:

```
BEGIN
WSN_setup;
WHILE (No_packets > 0)
  IF (path_found = false) & (D_coordinates = false)
    RREQ_area = WSN_area;
      REPEAT
        [path_found, path] = send_RREQ(S, RREQ_area);
        wait(T_rreq);
        No_RREQs = No_RREQs + 1;
      UNTIL (path_found = true) || (No_RREQ = limit)
  END IF
  IF (path_found = false) & (D_coordinates = true)
    [D_neighbour, Td, buffer] = estim(D);
    RREQ_area=area(S,D_neighbour);
      REPEAT
        [path_found, path] = send_RREQ(S, D_neighbour, RREQ_area);
        wait(T_rreq);
        No_RREQs = No_RREQs + 1;
      UNTIL (path_found = true) || (No_RREQ = limit)
```

```
END IF
IF (path_found = true)
   No_RREQs = 0;
   IF (buffer = true)
      REPEAT
         [path_found, buffer] = send_data(S, D_neighbour, Td, path);
         No_packets=No_packets-1;
         wait(Ts);
      UNTIL (No_packets = 0) || (path_found = false) || (Td = 0) || (buffer = false)
   ELSE
      REPEAT
         [path_found] = send_data(S, D, path);
         No_packets = No_packets-1;
         wait(Ts);
      UNTIL (No_packets = 0) || (path_found = false)
END IF
IF (No_RREQs = limit) || (Td = 0)
   No_RREQs=0;
   wait(T_retry);
   RREQ_area = WSN_area;
END IF
```

41. END WHILE

42. END

The explanation for each pseudocode line is given bellow:

1. Start the program.

2. "WSN_setup" is a function in which the following WSN parameters are defined: network area width and length, number of nodes, number of packets, D's speed, node transmission range, node transmit power, path loss exponent, the standard deviation for shadowing model, sensitivity threshold, route request packet size, route reply packet size, hello packet size, data packet size, data rate, energy per bit spent on transmission, energy per bit spent for reception, initial node energy.

3. The while syntax from 3 to 35 will search a path from S to D (or D next neighbour) and if a path is found will send data packets as long as "No_packets > 0" (where No_packets represent the total number of packets).

4. Checks if a path from S to D is not known and there is no knowledge about D coordinates.

5. The area in which RREQ are sent becomes the entire WSN.

6. RREQs are sent in the entire WSN each "T_rreq" seconds.

7. A RREQ is sent in the entire WSN, "path_found" becomes true when a path is found and the path from S to D is returned in "path".

8. Wait for "T_rreq" seconds between two RREQs.

9. The number of RREQs sent in the WSN are counted by "No_RREQ".

10. RREQs are sent until a path is found ("path_found = true") or a certain number of RREQs are sent (No_RREQ = limit).

11. It ends the case for which a path from S to D is searched when D coordinates are not known.

12. Checks if a path from S to D is not known and there is knowledge about D previous coordinates when it was connected with the WSN.

13. D next neighbour coordinates together with the disconnected time that D travels before entering WSN area are estimated based on D last known coordinates and direction of movement. Also “buffer” becomes true which means that the packets meant for D are sent to D next neighbour buffer.

14. The “RREQ_area” becomes the area determined by S coordinates and D next neighbour estimated coordinates.

15. RREQs are sent each “T_rreq” seconds.

16. A RREQ is sent in the “RREQ_area”, “path_found” becomes true when a path is found from S to D next neighbour, in “path” the path from S to D next neighbour is returned.

17. Wait for “T_rreq” seconds between two RREQs.

18. The number of RREQs sent in the WSN is counted by “No_RREQ”.

19. RREQs are sent until a path is found (“path_found = true”) or a certain number of RREQs are sent (No_RREQ = limit).

20. It ends the case for which a path from S to D next neighbour is searched when D last coordinates are known.

21. It starts the scenario in which data packets are sent if a path is found between S and D (or D next neighbour).

22. The number of RREQs sent in the WSN is set to 0.

23. Checks if the path is between S and D next neighbour.

24. S starts sending data packets to D next neighbour buffer.

25. Packets are sent from S to D next neighbour buffer. When D rejoins the WSN through its next neighbour, S will be informed to stop the data transition until all the buffered packets are sent to D and “buffer” becomes false.

26. “No_packets” represents the total number of packets which need to be sent.

27. Between two data packets transmission S waits for Ts seconds.

28. S sends data packets to D next neighbour if one of the following conditions is true:

- all the packets are sent (“No_packets = 0”).

- the path is broken (“path_found = false”).

- “Td = 0” , it is estimated that after Td seconds D will rejoin the network through its next neighbour and will inform S to pause the data transmission until all the buffered packets are received. However, in order to avoid the situation in which D will not rejoin the WSN through the estimated neighbour, S sends packets to D next neighbour buffer only for Td seconds.

- the buffer becomes false which means that D rejoined the network (“buffer = false”).

29. The path is between S and D.

30. S starts sending data packets to D.

31. Packets are sent from S to D and “path_found” is returned false when the path breaks (D exits the WSN area).

32. “No_packets” represents the total number of packets which need to be sent.

33. Between two data packets transmission S waits for Ts seconds.

34. S sends packets to D until all the packets are sent or D exits the WSN coverage area (path_found = false).

35. The scenario for which a path is known between S and D (or D next neighbour) and data packets are sent ends.

36. Checks if the maximum number of RREQs packets have been sent in the WSN or data packets have been sent to D's neighbour buffer for Td seconds.

37. No_RREQs becomes 0.

38. S waits for a certain period of time (T_retry seconds) after which it will restart sending RREQs to search a path to D.

39. The RREQ area becomes the entire WSN.

40. It ends the case when No_RREQs has reached the maximum value or Td seconds passed since S started to send data packets to D's neighbour buffer.

41. The scenario for which S has packets to send ends.

42. End program.

Bibliography

[1] Zhigang Wei Guangming Song, Yaoxin Zhou and Aiguo Song. A smart node architecture for adding mobility to wireless sensors networks. *Sensors and Actuators*, 147, Issue 1:216–221, 15 September 2008.

[2] http://www.purelink.ca/en/technologies/related technologies.php.

[3] K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE*, 11(6):54–61, 2004.

[4] http://www.embedded.arch.ethz.ch/uploads/Projects/WirelessSensorNode/nodeproto.jpg.

[5] Xudong Wang Ian F. Akyildiz. *Wireless Mesh Networks*. A John Wiley and Sons, Ltd, Publication, 2009.

[6] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Comput. Netw. ISDN Syst.*, 47(4):445–487, March 2005.

[7] M.P. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic. A survey of military applications of wireless sensor networks. In *Embedded Computing (MECO), 2012 Mediterranean Conference on*, pages 196–199, 2012.

[8] I.G. Tudorache, A.M. Popescu, and A.H. Kemp. Improved mesh wsn support for a realistic mobility model. In *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pages 340–344, 2010.

[9] J. Cheng S. Lu L. Zhang H. Luo, F. Ye. Ttdd: Two-tier data dissemination in large-scale wireless sensor networks. *Wireless networks*, 1:161–175, 11, January, 2005.

[10] A. Alemdar and M. Ibnkahla. Wireless sensor networks: Applications and challenges. In *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, pages 1–6, 2007.

[11] D. Puccinelli and M. Haenggi. Wireless sensor networks: applications and challenges of ubiquitous sensing. *Circuits and Systems Magazine, IEEE*, 5(3):19–31, 2005.

[12] J. Stankovic. When sensor and actuator networks cover the world. *ETRI Journal*, 30(5):627 • 633, 2008.

[13] U. Prathap, D.P. Shenoy, K.R. Venugopal, and L.M. Patnaik. Wireless sensor networks applications and routing protocols: Survey and research challenges. In *Cloud and Services Computing (ISCOS), 2012 International Symposium on*, pages 49–56, 2012.

[14] I.A. Essa. Ubiquitous sensing for smart and aware environments. *Personal Communications, IEEE*, 7(5):47–49, 2000.

[15] Sven Meyer and Andry Rakotonirainy. A survey of research on context-aware homes. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21*, ACSW Frontiers '03, pages 159–168, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[16] Waylon Brunette, Carl Hartung, Ben Nordstrom, and Gaetano Borriello. Proximity interactions between wireless sensors and their application. In *Proceed-*

ings of the 2nd ACM international conference on Wireless sensor networks and applications, WSNA '03, pages 30–37, New York, NY, USA, 2003. ACM.

[17] Matthias Gauger, Daniel Minder, Pedro José Marrón, Arno Wacker, and Andreas Lachenmann. Prototyping sensor-actuator networks for home automation. In *Proceedings of the workshop on Real-world wireless sensor networks*, REALWSN '08, pages 56–60, New York, NY, USA, 2008. ACM.

[18] Verena Weber. Smart sensor networks: Technologies and applications for green growth. Technical report, Organisation for Economic Co-operation and Development Report, 2009.

[19] D. Estrin L. Girod M. Hamilton A. Cerpa, J. Elson and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Josi, Costa Rica*, 2001.

[20] K. Bridges E. Biagioni. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16:315–324, 2002.

[21] http://www.permasense.ch/.

[22] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGARCH Comput. Archit. News*, 30(5):96–107, October 2002.

[23] http://www.bannerengineering.com/.

[24] C. E. Perkins. *Ad Hoc Networking.* Addison Wesley, 2001.

[25] Ch. Antonopoulos S. Giannoulis T. Stoyanova G. Papadopoulos A. Prayati, F. Kerasiotis. An overview of development problems in wsns. In *Interna-*

tional Workshop on Wireless Pervasive Healthcare (WiPH 2009), London UK., March 2009.

[26] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, March 2002.

[27] G. Hoblos, M. Staroswiecki, and A. Aitouche. Optimal design of fault tolerant sensor networks. In *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, pages 467–472, 2000.

[28] J. Fadlullah and Mohsen Kavehrad. Indoor high-bandwidth optical wireless links for sensor networks. *Lightwave Technology, Journal of*, 28(21):3086–3094, 2010.

[29] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 217–229, New York, NY, USA, 2003. ACM.

[30] B. Malarkodi, P. Gopal, and B. Venkataramani. Performance evaluation of adhoc networks with different multicast routing protocols and mobility models. In *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, pages 81–84, 2009.

[31] A. Kumar, R. Sood, H. Saran, and R. Shorey. Performance of multiple tcp connections over different routing protocols in mobile ad-hoc networks. In *Personal Wireless Communications, 2000 IEEE International Conference on*, pages 500–503, 2000.

[32] Xiaojiang Du. A simulation study of an energy efficient routing protocol for mobile ad hoc networks. In *Simulation Symposium, 2004. Proceedings. 37th Annual*, pages 125–131, 2004.

[33] Hua-Wen Tsai, Tzung-Shi Chen, and Chih-Ping Chu. An on-demand routing protocol with backtracking for mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 3, pages 1557–1562 Vol.3, 2004.

[34] Zhao Chenchen and Yang Zhen. A new eaaodv routing protocol based on mobile agent. In *Systems and Networks Communications, 2006. ICSNC '06. International Conference on*, pages 4–4, 2006.

[35] I.G. Tudorache, A.M. Popescu, and A.H. Kemp. Manet routing protocols problem for the marginal mobility model. In *Microwave Conference (EuMC), 2011 41st European*, pages 139–142, 2011.

[36] Bo Peng A.H. Kemp Ana Maria Popescu, Ion Gabriel Tudorache. Surveying position based routing protocols for wireless sensor and ad-hoc networks. *International Journal of Communication Networks and Information Security (IJCNIS)*, Vol. 4, No. 1:41–67, April 2012.

[37] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, July 2000.

[38] David B. Johnson, David A. Maltz, and Josh Broch. Ad hoc networking. chapter DSR: the dynamic source routing protocol for multihop wireless ad hoc networks, pages 139–172. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[39] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 1999.

[40] Ha Duyen Trung, Watit Benjapolakul, and Phan Minh Duc. Performance evaluation and comparison of different ad hoc routing protocols. *Computer*

Communications, Special issue on security on wireless ad hoc and sensor networks, 30(11-12):2478 – 2496, 2007.

[41] Shapina Abdullah Siti Rahayu Abdul Aziz, Nor Adora Endut and Mior Norazman Mior Daud. Performance evaluation of aodv,dsr and dymo routing protocol in manet. In *Conference on Scientific and Social Research*, 14-15 March 2009.

[42] E. Ahvar and M. Fathy. Performance evaluation of routing protocols for high density ad hoc networks based on energy consumption by glomosim simulator, 2007.

[43] Hyunwoo Kang and Dongkyun Kim. Vector routing for delay tolerant networks. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5, 2008.

[44] YoungSang Yun and Y. Xia. Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications. *Mobile Computing, IEEE Transactions on*, 9(9):1308–1318, 2010.

[45] Jun Luo and J-P Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1735–1746 vol. 3, 2005.

[46] E.M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2):46–55, 1999.

[47] F. Bai and A. Helmy. *Book Chapter 1 in Wireless Ad Hoc and Sensor Networks.* Kluwer academic, 2004.

[48] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta

Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '98, pages 85–97, New York, NY, USA, 1998. ACM.

[49] A.A. Agashe and S.K. Bodhe. Performance evaluation of mobility models for wireless ad hoc networks. In *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on*, pages 172–175, 2008.

[50] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *WIRELESS COMMUNICATIONS & MOBILE COMPUTING (WCMC): SPECIAL ISSUE ON MOBILE AD HOC NETWORKING: RESEARCH, TRENDS AND APPLICATIONS*, 2:483–502, 2002.

[51] Chris Barrett, Achla Marathe, Madhav V. Marathe, and Martin Drozda. Characterizing the interaction between routing and mac protocols in ad-hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 92–103, New York, NY, USA, 2002. ACM.

[52] Fan Bai, N. Sadagopan, and A. Helmy. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 825–835 vol.2, March 2003.

[53] Mohamed Younis Kemal Akkaya. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349, May 2005.

[54] S. Pal, D. Bhattacharyya, G.S. Tomar, and Tai hoon Kim. Wireless sensor

networks and its routing protocols: A comparative study. In *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*, pages 314–319, 2010.

[55] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM MOBILE COMPUTING AND COMMUNICATIONS REVIEW*, 6:28–36, 2002.

[56] W. Heinzelman. Application specific protocol architectures for wireless networks, phd thesis, mit, 2000.

[57] *B. Krishnamachari, D. Estrin, S. Wicker, Modeling data centric routing in wireless sensor networks, in: Proceedings of IEEE INFOCOM, New York*, June 2002.

[58] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, 2000.

[59] *L. Subramanian, R.H. Katz, An architecture for building self configurable systems, in: Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA.*, August 2000.

[60] Silvia Giordano, Ivan Stojmenovic, and Ljubica Blazevic. Position based routing algorithms for ad hoc networks: A taxonomy. In *Ad Hoc Wireless Networking*, pages 103–136. Kluwer, 2001.

[61] I. Stojmenovic. Position-based routing in ad hoc networks. *Communications Magazine, IEEE*, 40(7):128–134, 2002.

[62] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with

guaranteed delivery in ad hoc wireless networks. *Wirel. Netw.*, 7(6):609–616, November 2001.

[63] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *Communications, IEEE Transactions on*, 32(3):246–257, 1984.

[64] Cong Liu and Jie Wu. Swing: Small world iterative navigation greedy routing protocol in manets. In *Computer Communications and Networks, 2006. ICCCN 2006. Proceedings.15th International Conference on*, pages 339–350, 2006.

[65] M. Rahman, M. Mambo, A. Inomata, and E. Okamoto. An anonymous on-demand position-based routing in mobile ad hoc networks. In *Applications and the Internet, 2006. SAINT 2006. International Symposium on*, pages 7 pp.–306, 2006.

[66] Wolfgang Kie'i, Holger F'o'iler, Jfrg Widmer, and Martin Mauve. Hierarchical location service for mobile ad-hoc networks. Technical report, 2003.

[67] J.Y. Le Boudec L. Blazevic, S.Giordano. Self organized terminode routing, cluster computing, 2002. pp. 215-218.

[68] S. Son B. Blum, T. He and J. Stankovic. Igf: A state-free robust communication protocol for wireless sensor networks. Technical Report CS-2013-11, Dept.of Computer Science, University of Virginia, 2003.

[69] B. Nath D. Niculescu. Trajectory based forwarding and its applications. MOBICOM, Sept. 2003. pp.14-19.

[70] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *Proceedings of the 2nd international conference on Em-*

bedded networked sensor systems, SenSys '04, pages 108–121, New York, NY, USA, 2004. ACM.

[71] I. Stojmenovic and Xu Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 12(10):1023–1032, 2001.

[72] R. Jain, A. Puri, and Raja Sengupta. Geographical routing using partial information for wireless ad hoc networks. *Personal Communications, IEEE*, 8(1):48–57, 2001.

[73] Marc Heissenb'ottel, Torsten Braun, Thomas Bernoulli, and Markus Wdchli. Blr: beacon-less routing algorithm for mobile ad hoc networks. *Computer Communications*, 27(11):1076 – 1086, 2004. <ce:title>Applications and Services in Wireless Networks</ce:title>.

[74] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 243–254, New York, NY, USA, 2000. ACM.

[75] Hannes Frey and Ivan Stojmenovic. On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, MobiCom '06, pages 390–401, New York, NY, USA, 2006. ACM.

[76] JuanA. Sanchez and PedroM. Ruiz. Locally optimal source routing for energy-efficient geographic routing. *Wireless Networks*, 15(4):513–523, 2009.

[77] M. Strasser A. Weibel A survey on routing metrics TIK-Report 2007 R. Baumann, S. Heimlicher.

[78] M.Younis K.Akkaya. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 1:325–349, May, 2005.

[79] C. Lino, C.T. Calafate, A. Diaz-Ramirez, P. Manzoni, and J.-C. Cano. An efficient solution offering sink mobility support in wireless sensor networks. In *Wireless Telecommunications Symposium (WTS), 2012*, pages 1–9, 2012.

[80] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, New York, NY, USA, 2009. ACM.

[81] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, February 2003.

[82] Boon Thau Loo, Joseph M. Hellerstein, Ion Stoica, and Raghu Ramakrishnan. Declarative routing: Extensible routing with declarative queries. *SIGCOMM Comput. Commun. Rev.*, 35(4):289–300, August 2005.

[83] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. *Wirel. Netw.*, 11(3):285–298, May 2005.

[84] Michal Piorkowski Matthias Grossglauser Jean-pierre Hubaux Jun Luo, Jacques Panchard. Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. *Sensor Networks*, 2nd IEEE/ACM DCOSS, 2006.

[85] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (dream). In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Comput-*

ing and Networking, MobiCom '98, pages 76–84, New York, NY, USA, 1998. ACM.

[86] Li-Minn; Seng Kah-Phooi Zungeru, A. M.; Ang. Termite-hill: Routing towards a mobile sink for improving network lifetime in wireless sensor networks. In *Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 622,627, 8-10 Feb. 2012.

[87] Culler D. Woo A., Tong T. Taming the underlying challenges of reliable mulmulti routing in sensor networks. In *1st ACM SenSys.*, 2003.

[88] Hyung Seok Kim, Tarek F. Abdelzaher, and Wook Hyun Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 193–204, New York, NY, USA, 2003. ACM.

[89] C. Konstantopoulos, G. Pantziou, Damianos Gavalas, A. Mpitziopoulos, and B. Mamalis. A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(5):809–817, May 2012.

[90] Jayanthi Rao and Subir Biswas. Network-assisted sink navigation for distributed data gathering: Stability and delay-energy trade-offs. *Comput. Commun.*, 33(2):160–175, February 2010.

[91] Dongjin Son, A. Helmy, and B. Krishnamachari. The effect of mobility-induced location errors on geographic routing in mobile ad hoc sensor networks: analysis and improvement using mobility prediction. *Mobile Computing, IEEE Transactions on*, 3(3):233–245, 2004.

[92] T. Melodia, D. Pompili, and I.F. Akyldiz. Handling mobility in wireless sensor

and actor networks. *Mobile Computing, IEEE Transactions on*, 9(2):160–173, Feb 2010.

[93] S.H. Shah and K. Nahrstedt. Predictive location-based qos routing in mobile ad hoc networks. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 2, pages 1022–1027 vol.2, 2002.

[94] Min-Chieh Yu and Jenq-Shiou Leu. Kernel weighted scheme for improving mobile sensor-node connectivity. *Sensors Journal, IEEE*, 13(4):1200–1206, April 2013.

[95] Euisin Lee, Soochang Park, Donghun Lee, Younghwan Choi, Fucai Yu, and Sang-Ha Kim. A predictable mobility-based communication paradigm for wireless sensor networks. In *Communications, 2007. APCC 2007. Asia-Pacific Conference on*, pages 373–376, Oct 2007.

[96] A. Munari, W. Schott, and S. Krishnan. Energy-efficient routing in mobile wireless sensor networks using mobility prediction. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 514–521, Oct 2009.

[97] William Su, Sung-Ju Lee, and Mario Gerla. Mobility prediction and routing in ad hoc wireless networks. *Int. J. Netw. Manag.*, 11(1):3–30, January 2001.

[98] H. Wu, Yu Wang, Ha Dang, and Feng Lin. Analytic, simulation, and empirical evaluation of delay/fault-tolerant mobile sensor networks. *Wireless Communications, IEEE Transactions on*, 6(9):3287–3296, 2007.

[99] A.C.B.K. Vendramin, A. Munaretto, M.R. Delgado, and A.C. Viana. A greedy ant colony optimization for routing in delay tolerant networks. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 1127–1132, 2011.

[100] Fulong Xu, Ming Liu, Jiannong Cao, Guihai Chen, Haigang Gong, and Jinqi Zhu. A motion tendency-based adaptive data delivery scheme for delay toler-

ant mobile sensor networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, 2009.

[101] R.S. Mangrulkar and M. Atique. Routing protocol for delay tolerant network: A survey and comparison. In *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, pages 210–215, 2010.

[102] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *SIGPLAN Not.*, 37(10):96–107, October 2002.

[103] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[104] R.C. Shah, S. Roy, S. Jain, and W. Brunette. 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*, pages 30–41, May 2003.

[105] YoungSang Yun, Y. Xia, B. Behdani, and J.C. Smith. Distributed algorithm for lifetime maximization in delay-tolerant wireless sensor network with mobile sink. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 370–375, Dec 2010.

[106] Hyunwoo Kang and Dongkyun Kim. Vector routing for delay tolerant networks. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5, Sept 2008.

[107] Fulong Xu, Ming Liu, Jiannong Cao, Guihai Chen, Haigang Gong, and Jinqi

Zhu. A motion tendency-based adaptive data delivery scheme for delay tolerant mobile sensor networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, Nov 2009.

[108] Jinqi Zhu, Jiannong Cao, Ming Liu, Yuan Zheng, Haigang Gong, and Guihai Chen. A mobility prediction-based adaptive data gathering protocol for delay tolerant mobile sensor network. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5, Nov 2008.

[109] Xing Xu, Ji Luo, and Qian Zhang. Delay tolerant event collection in sensor networks with mobile sink. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.

[110] H. Wu, Yu Wang, Ha Dang, and Feng Lin. Analytic, simulation, and empirical evaluation of delay/fault-tolerant mobile sensor networks. *Wireless Communications, IEEE Transactions on*, 6(9):3287–3296, September 2007.

[111] YoungSang Yun and Y. Xia. Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications. *Mobile Computing, IEEE Transactions on*, 9(9):1308–1318, Sept 2010.

[112] A. Nayebi, H. Sarbazi-Azad, and G. Karlsson. Routing, data gathering, and neighbor discovery in delay-tolerant wireless sensor networks. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–6, 2009.

[113] BaiJun Wu, Feng Lin, and Jiliu Zhou. Adaptive routing in delay-tolerant mobile sensor network. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010 International Conference on*, pages 79–86, 2010.

[114] N. Ng, Hwa Chang, Zhongjian Zou, and Sai Tang. An adaptive threshold

method to address routing issues in delay-tolerant networks. In *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pages 1122–1126, 2011.

[115] http://www.isi.edu/nsnam/ns/.

[116] http://www.scalable networks.com/products/qualnet/.

[117] http://www.mathworks.com/.

[118] Shaila K K R Venugopal L M Patnaik. S H Manjula, C N Abhilash. Performance of aodv routing protocol using group and entity mobility models in wireless sensor networks. In *In Proceedings of the International MultiConference of Engineers and Computer Scientists,*, 19-21 March 2008, Hong Kong.

[119] Canfeng Chen and Jian Ma. Simulation study of aodv performance over ieee 802.15.4 mac in wsn with mobile sinks. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 2, pages 159–164, May 2007.

[120] M.N. Jambli, K. Zen, H. Lenando, and A. Tully. Performance evaluation of aodv routing protocol for mobile wireless sensor network. In *Information Technology in Asia (CITA 11), 2011 7th International Conference on*, pages 1–6, July 2011.

[121] M. Pandey and S. Verma. Performance evaluation of aodv for different mobility conditions in wsn. In *Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011 International Conference on*, pages 240–243, Dec 2011.

[122] Imran Aali Jokhio Javed Ali Baloch and Sana Hoor Jokhio. Performance evaluation of spatial vector routing protocol for wireless sensor networks. *Mehran University Research Journal of Engineering & Technology,*, Volume 31, No. 4,, October, 2012.

[123] Salwa El Ramly Magdy M. Ibrahim Adel Gaafar A.Elrahim, Hussein A.Elsayed. An energy aware wsn geographic routing protocol. *Universal Journal of Computer Science and Engineering Technology*, pages 105–111, Nov. 2010.

[124] Tian He, J.A. Stankovic, Chenyang Lu, and T. Abdelzaher. Speed: a stateless protocol for real-time communication in sensor networks. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, pages 46–55, May 2003.

[125] Saquib Razak, Vinay Kolar, Nael B. Abu-Ghazaleh, and Khaled A. Harras. How do wireless chains behave?: the impact of mac interactions. In *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '09, pages 212–220, New York, NY, USA, 2009. ACM.

[126] Y.B. Ko and N.H.: Vaidya. Location aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6:307–321, 2000.

[127] Salman N. Kemp A.H. Rasool, I. Rssi-based positioning in unknown path-loss model for wsn. In *Sensor Signal Processing for Defence (SSPD), London, UK*, 25-27 Sept. 2010.

[128] A. M. Popescu, G. I. Tudorache, and A.H. Kemp. Performance study of node placement for geographic routing in wsns. In *Communication Technologies Workshop (Swe-CTW), 2011 IEEE Swedish*, pages 13–18, 2011.

[129] A. M. Popescu, N. Salman, and A.H. Kemp. Energy consumption of geographic routing with realistic localisation. *Networks, IET*, 1(3):126–135, 2012.

[130] A. M. Popescu, N. Salman, and A. H. Kemp. Energy consumption analysis of

geographic routing in wsns with location error. In *European Wireless, 2012. EW. 18th European Wireless Conference*, pages 1–8, 2012.

[131] B. Peng, R. Mautz, A.H. Kemp, W. Ochieng, and Q. Zeng. On the effect of localization errors on geographic routing in sensor networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 3136–3140, May 2008.

[132] R.C. Shah, A. Wolisz, and J.M. Rabaey. On the performance of geographical routing in the presence of localization errors [ad hoc network applications]. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 5, pages 2979–2985 Vol. 5, May 2005.

[133] B. Peng and A.H. Kemp. Impact of location errors on geographic routing in realistic wsns. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–7, Sept 2010.

[134] You-Chiun Wang, Che-Hsi Chuang, Yu-Chee Tseng, and Chien-Chung Shen. A lightweight, self-adaptive lock gate designation scheme for data collection in long-thin wireless sensor networks. *Wireless Communications and Mobile Computing*, 13(1):47–62, 2013.

[135] Yongjin Kim, Jae-Joon Lee, and Ahmed Helmy. Modeling and analyzing the impact of location inconsistencies on geographic routing in wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(1):48–60, January 2004.

[136] Jayanthi Rao and Subir Biswas. Network-assisted sink navigation for distributed data gathering: Stability and delay-energy trade-offs. *Computer Communications*, 33(2):160 – 175, 2010.

[137] Tommaso Melodia, Dario Pompili, and Ian F. Akyildiz. Handling mobility in

wireless sensor and actor networks. *IEEE Transactions on Mobile Computing*, 9(2):160–173, 2010.

[138] A. Munari, W. Schott, and S. Krishnan. Energy-efficient routing in mobile wireless sensor networks using mobility prediction. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 514–521, 2009.