ENERGY EFFICIENT COORDINATE ESTABLISHMENT IN WIRELESS SENSOR NETWORKS

by

Daniel Jacobus Elardus Erasmus

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the
Faculty of Engineering, Built Environment and Information Technology
UNIVERSITY OF PRETORIA
Pretoria

August 2005

DISSERTATION SUMMARY

ENERGY EFFICIENT COORDINATE ESTABLISHMENT IN WIRELESS SENSOR NETWORKS

by

Daniel Jacobus Elardus Erasmus

Supervisor:	Prof. G. P. Hancke
Department:	Electrical, Electronic and Computer Engineering
	UNIVERSITY OF PRETORIA
Degree:	M.Eng (Computer Engineering)

Wireless Sensor Networks (WSNs) refer to a group of spatially deployed devices which are used to monitor or detect phenomena, and have the ability to relay sensed data and signalling wirelessly.

Positioning information in WSNs is absolutely crucial to perform tasks such as intelligent routing, data aggregation and data collection optimally. A need exists for localisation algorithms which are scalable, distributed, energy efficient and easy to deploy.

This research proposes a beaconless Cluster-based Radial Coordinate Establishment (CRCE) positioning algorithm to locate sensor nodes relative to a local coordinate system. The system does not make use of Global Positioning System (GPS) or any other method to provide a-priori position information for a set of nodes prior to the CRCE process.

The objective of CRCE is to reduce energy consumption while providing a scalable coordinate establishment method for use in WSNs.

To reduce energy consumption during the node positioning process, the research focuses on minimising the number of message exchanges in the network by implementing a cluster-based network topology and utilising the potential of geographically distributed processors.

A *radial* coordinate convergence process is proposed to achieve scalability as the number of sensors in the network increases.

Three other localisation algorithms are investigated and compared to CRCE to identify the one best suited for coordinate establishment in WSNs. Two of these comparison algorithms are published in the literature and the other is a modified version of one of the published algorithms.

The results show a significant decrease in the number of messages that are necessary to establish a network-wide coordinate system successfully, ultimately making it more scalable and energy efficient. In addition, position based algorithms, such as location based routing, can be deployed on top of CRCE.

Keywords:

- Beaconless
- Cluster-based
- Clusterless
- Coordinate establishment
- Energy efficient
- Iterative convergence
- Localisation
- Positioning
- Radial convergence
- Wireless Sensor Networks

SAMEVATTING VAN VERHANDELING

ENERGIE-DOELTREFFENDE KOÖRDINAATOPRIGTING IN DRAADLOSE SENSORNETWERKE

deur

Daniel Jacobus Elardus Erasmus

Toesighouer:	Prof. G. P. Hancke
Departement:	Elektriese, Elektroniese en Rekenaar-Ingenieurswese UNIVERSITEIT VAN PRTORIA
Graad:	M.Ing (Rekenaar Ingenieurswese)

Draadlose sensornetwerke (DSNe) verwys na ‘n geografiesverspreide groep apparate wat gebruik word om ‘n verskynsel te gewaar of te monitor en het die vermoë om waargenome data en beheerseine draadloos te versend.

Inligting betreffende node posisies in DSNe is absoluut van die uiterste belang om take soos die intelligente aanstuur van boodskappe, data-samevoeging en data-insameling optimaal te kan uit voer. Daar bestaan ‘n behoefte vir lokalisasie algoritmes wat skaalbaar, verspreid, energiedoeltreffend en maklik ontplooibaar is.

In hierdie navorsing word ‘n bakenlose klustergebasseerde radiale koördinaatoprigting (KRKO) posisioneringsalgoritme voorgestel om posisies van sensornodes vas te stel, relatief tot ‘n lokale koördinaatstelsel. Die sisteem maak nie van GPS of enige ander metode gebruik om posisie-informasie vooraf vir ‘n stel nodes beskikbaar te stel nie.

Die doel van KRKO is om energieverbruik te verminder, terwyl dit ‘n skaleerbare koördinaatoprigtingsmetode in DSNs voorsien.

Om energieverbruik gedurende die posisioneringsproses te verminder, fokus die navorsing op die minimalisering van die aantal boodskapuitruilings in die netwerk. Dit word moontlik gemaak d.m.v. ‘n klustergebasseerde netwerktopologie en die potensiaal van geografiesverspreide verwerkers.

‘n Radiale koördinaatomskakelingsprosedure word voorgestel om skaleerbaarheid te verkry soos wat die aantal sensors in ‘n netwerk vermeerder.

Drie ander posisioneringsalgoritmes word ondersoek en word vergelyk met KRKO om die bes moontlike koördinaatoprigtingsmetode te identifiseer. Twee van hierdie vergelykings-algoritmes is in die literatuur gepubliseer en die ander een is ‘n veranderde weergawe van een van die gepubliseerde algoritmes.

Die resultate toon ‘n beduidende vermindering in die aantal boodskappe wat nodig is om ‘n netwerkwye koördinaatstelsel suksesvol op te rig. Dit maak die stelsel skaleerbaar en energie-doeltreffend. Posisiegebasseerde algoritmes, soos boodskapaanstuur algoritmes wat posisie as aanstuurmaatstaf gebruik, kan ook bo-op KRKO ontplooi word.

Sleutelwoorde:

- Bakenloos
- Draadlose Sensornetwerke
- Energie-doeltreffendheid
- Herhalende omskakeling
- Klustergebasseerd
- Klusterloos
- Koördinaatoprigting
- Lokalisasie
- Positionering
- Radiale omskakeling

LIST OF ABBREVIATIONS

AHLoS	Ad-hoc Localization System
AOA	Angle of Arrival
APS	Ad-hoc Positioning System
BS	Base Station
CH	Cluster Head
C-less	Cluster-less
CRCE	Cluster-based Radial Coordinate Establishment
CSMA-MPS	Carrier Sense Multiple Access – Minimum Preamble Sampling
DV	Distance Vector
GPS	Global Positioning System
LOS	Line Of Sight
LPS	Local Positioning System
LRG	Location Reference Group
MAC	Medium Access Control
MDS	Multidimensional Scaling
MF	Mobility Framework
OMNeT++	Objective Modular Network Testbed in C++
PHY	Physical layer
POS	Personal Operating Space
RC-less	Radial and Cluster-less
R-less	Radial-less
RSS	Received Signal Strength
RTOF	Roundtrip Time Of Flight
SNR	Signal-to-Noise Ratio
SPA	Self Positioning Algorithm
TDOA	Time Difference Of Arrival
WSAN	Wireless Sensor and Actor Network
WSN	Wireless Sensor Network

TABLE OF CONTENTS

CHAPTER 1 RESEARCH OVERVIEW

1.1 Introduction

Wireless Sensor Networks (WSNs) are becoming an increasingly attractive means to get insight into the behaviour and characteristics of modern day dynamic systems. The availability of energy efficient embedded processors, radios, low-power micro-sensors and actuators, is enabling the application of *distributed wireless sensing* to a wide range of applications. Some of these applications include environmental monitoring, smart spaces, medical applications, and precision agriculture [1]. Valuable data are collected, integrated, and utilized to enhance production processes, increase revenue, and decrease security risks, to name but a few.

Wired sensor networks have been around for many years, with an array of gauges measuring anything from temperature, fluid levels and humidity to other attributes on pipelines, pumps, generators and manufacturing lines [2]. Most of these run as separate wired networks, often linked to a computer or a control panel that has a warning system – flashes lights or sounds an alarm when a temperature rises too high or a machine vibrates too much. Actuators are also included in the setup. These let the control panel slow down a pump or turn on a heater or a fan in response to the sensor data.

- *Why wireless?*

  When a network of distributed sensors can be easily achieved and wired to renewable (relatively infinite) energy sources, it is often the more advantageous approach. This greatly simplifies the system design, signalling and operation [1]. However, in many current and envisioned applications, the environment being monitored does not have installed infrastructure for either communications or energy. Therefore the nodes must rely on local, finite, and relatively small energy sources and wireless communication channels.

  WSNs eliminate wires, and their installation and maintenance costs. The routing algorithms will also enable these wireless networks to route around nodes that fail or whose radio signal is hammered by interference from heavy equipment.

- *Why distributed sensing?*

  Often the precise location of a signal of interest is unknown in a monitored region. Distributed sensing allows one to deploy the sensors closer to the monitored phenomena than the case of only a single sensor being used. This yields higher SNR, and improved opportunities for line of sight.

  SNR can be addressed in many cases by deploying one very large sensitive sensor, but line of sight, and more generally obstructions, cannot be compensated for by deploying one sensor – regardless of its sensitivity. Therefore, distributed sensing provides robustness to environmental obstacles.

- *Is node positioning in WSNs necessary?*

  The need for positioning in WSNs depends on the type of application envisioned. For ad hoc or sensor networks used by the military in rescue applications, for example, positioning is indispensable. Other applications in which positioning plays a major role are applications such as meteorological and environmental monitoring, library archiving and package tracking [3].

  Localisation of nodes can either be the mean or the goal of a sensing application and enables more efficient protocols as well as a number of new applications. *Answer*: Yes.

1.2 Problem statement

Positioning information in WSNs is absolutely crucial to perform tasks such as intelligent routing, data aggregation and data collection optimally.

WSNs do not exist without their challenges. It can consist of thousands and even millions of resource constrained, tiny wireless sensor nodes. Therefore, *scalability* is a critical factor in the system design. *Energy efficiency* is another major concern in WSNs. Sensor nodes are very small, low cost devices and do not provide a means for recharging batteries or installing large power supplies.

The above mentioned makes it clear that there exists a need for a highly scalable, energy efficient localisation method.

1.3 Research objective

In this research we address all three of these key issues in the coordinate establishment stage of the network (right after node deployment), by proposing a distributed Cluster-based Radial Coordinate Establishment (CRCE) method.

The objective of this research is threefold:

- to develop a positioning method with the ability to establish a network-wide *relative coordinate system*,
- to reduce the energy consumption during the localisation process, and
- to increase the scalability of current similar coordinate establishment methods.

1.4 Scope

CRCE is a node coordinate establishment method for WSNs, and is located at layer 3 of the OSI model, the network layer – Figure 1.1.

Higher layers	Layers 4 - 7
Network	CRCE
Data link	Layer 2
Physical	Layer 1

Figure 1.1 CRCE in the OSI model

Medium access, transceive timing, and synchronization issues are all addressed by the lower layers and do not fall within the scope of this research. Nevertheless, the type of MAC layer has to be considered to be able to calculate the *message cost*[1]. This is done in Section 3.3.

In addition, node position refinement can be considered as a different research subject and is not part of the simulation and test scenarios.

CRCE assumes accurate distance measurements without loss of generality or implementation possibility, but knowledge of range measurement techniques/principles is necessary to understand the coordinate establishment process. To this end, measurement principles are discussed briefly in Section 2.3.

Figure 1.2 shows where sensor node positioning fits in the WSN life-cycle. Routing is clearly a separate entity from coordinate establishment, but once the CRCE process is completed, any location-based routing algorithm can be deployed on top of it.

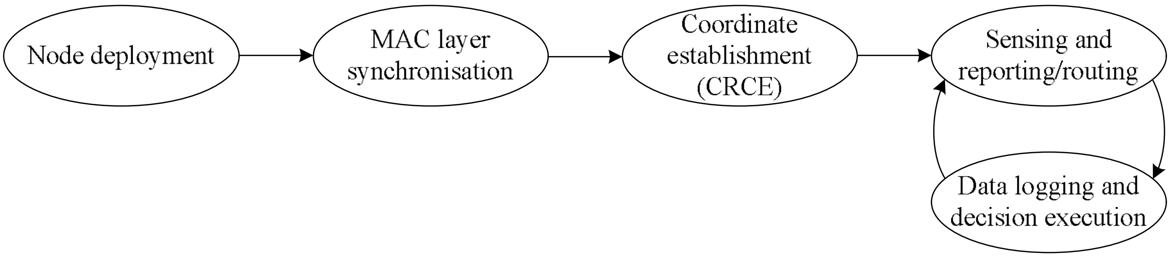


Figure 1.2 Coordinate establishment in the network life-cycle

1.5 Research approach

A hypothesis for localisation in WSNs is stated in this section. Some research questions are also presented together with the research approach that was followed in an attempt to answer these questions.

---

[1] The *message cost* of a localisation method refers to the total number of message transmissions necessary to establish a network-wide coordinate system.

1.5.1 Hypothesis formulation

A hypothesis was formulated regarding positioning in WSNs: “Coordinate establishment in WSNs is more expensive, in terms of message cost, when the network topology is clusterless and coordinate propagation is done in an iterative manner, as opposed to a cluster-based, radial coordinate convergence method”.

The hypothesis guided the research in an attempt to accept or reject it.

1.5.2 Research questions

Some questions were identified to understand the problem better and provide guidance to accomplish the set goals. These are:

- What is the current state of coordinate establishment in WSNs?
- What is the message cost of localisation methods?
- Is a reduction in message cost a viable way to reduce the consumed energy of a sensor?
- Can message cost in a network be calculated by a numerical formula?
- Can the message cost of current localisation methods be reduced?
- If so, is the reduction significant or not?
- Can the scalability of current localisation methods be increased?
- If so, what needs to be done to achieve this?
- Is relative localisation a viable solution for node positioning?
- Is radial coordinate propagation cheaper than its iterative counterpart?
- In what way does radial coordinate propagation influence networks with a cluster-based topology?
- In what way does radial coordinate propagation influence networks with a clusterless topology?
- How are node coordinates computed?

1.5.3 Research instruments

i. Literature study
    a. A scope for the research was defined.

b. A problem statement was formulated.
c. A hypothesis was formulated.
d. The trade-off between centralised and distributed processing was investigated.
e. Current relative localisation methods were studied.
f. Various ranging methods were investigated to understand the principles of node positioning.
g. Comparison methods for CRCE was identified.

ii. CRCE design
    a. An algorithm for node localisation was designed, called Cluster-based Radial Coordinate Establishment, or CRCE.
    b. As the name suggests, the coordinate updates traverse the network radially, rather than iteratively.
    c. A cluster-based approach was followed in the network topology.

iii. Implementation
    a. The CRCE algorithm was programmed in an appropriate programming language.
    b. CRCE was implemented in a suitable simulation environment.
    c. It was also analysed numerically.
    d. Equations for message cost were obtained.

iv. Assessment
    a. CRCE was compared to three other localisation methods:
        - iterative and clusterless,
        - radial and clusterless, and
        - iterative and cluster-based.
    b. The three algorithms were also simulated and numerically analysed.
    c. These were compared in terms of message cost.
    d. Results are provided and discussed.
    e. A final conclusion is given.

1.6 Chapter summary

The first part of this chapter ascertained the necessity of position enabled wireless sensor networks.

A problem statement was provided and established the need for an energy efficient and scalable localisation method. In order to address this, the research objectives were stated and the CRCE method was proposed.

The scope of the research was set to clearly mark the span of the required work and keep the focus on the end goals. Finally, the research approach was given (including some research questions and a hypothesis) to understand the problem better and to guide the efforts toward these set goals/objectives.

CHAPTER 2 LITERATURE STUDY

The current industry standards relating to the field of Wireless Sensor Networks are briefly discussed in Section 2.1. After that, a short section on the background of WSNs serves as an introduction for the rest of this chapter.

The aspects that are considered in Chapter 2 are geared towards localisation issues in WSNs. These include: range measurement and processing principles in WSNs, relative and one-hop positioning methods, and a quick view on location-based routing algorithms.

2.1 Current standards

The IEEE Std 802.15.4 was approved in May 2003. The purpose of the document is to "*provide a standard for ultra-low complexity, ultra-low cost, ultra-low power consumption, and low data rate wireless connectivity among inexpensive devices*" [4].

The 802.15.4 project also intended to work towards a level of coexistence with other wireless devices in conjunction with Coexistence Task Groups, such as 802.15.2 and 802.11™/ETSI-BRAN/MMAC 5GSG [5] – [8].

The scope of the standard is restricted to the physical layer (PHY) and medium access control (MAC) layer. The document defines the specifications of these two layers for low data rate wireless connectivity with fixed, portable, and moving devices in mind. In addition, these devices are defined to have no battery or very limited battery consumption requirements, typically operating in the personal operating space (POS) of 10 m [4].

The ZigBee Alliance was subsequently formed to address issues relating to the higher layers of a wireless sensor network [9]. It is a non-profit industry consortium of semiconductor manufacturers, technology providers, OEMs and end-users worldwide, with a current membership of 150+.

The aim of the alliance is to create a specification which defines mesh, peer-to-peer, and cluster tree network topologies with data security features and interoperable application profiles. The specification is based on the IEEE 802.15.4 standard and relates to it in the same way as the Wi-Fi Alliance to the IEEE 802.11, [10] and [5] respectively.

2.2 WSN Background

A sensor network consists of numerous spatially distributed sensors called nodes. These are used to monitor or detect a variety of phenomena at different locations, such as temperature changes, vibration, pressure, movement or pollutant levels [11].

Sensors are intended to be physically small and inexpensive devices, making it possible to produce and deploy them in large numbers. A sensor node[2] consists of:

- a radio transceiver,
- a low power microcontroller,
- sensing device(s),
- a very limited finite energy source (e.g. a battery), and
- various software algorithms and applications.

Resources in terms of energy, memory, computational speed and bandwidth are severely constrained to meet the objective of the sensors being small and inexpensive.

Energy efficient location awareness in Wireless Sensor Networks is far from a trivial task. Any number of resource constrained sensors can be deployed in an ad-hoc fashion with no a-priori knowledge of their positions. The “traditional” method of adding GPS to a device that needs to know its location can no longer be incorporated. The reasons are [12]:

- *increased power consumption*: GPS will reduce, if not zero, the lifetime of node batteries for very small state-of-the-art sensor nodes,
- *large form factor*: it limits the reduction in node size,

---

[2] Throughout this document the terms: sensor, node, and sensor node are used interchangeably; all referring to the same entity, defined in this paragraph.

- *poor indoor reception*: walls and other obstacles can block the line-of-sight between the GPS enabled node and the satellite, and
- *high production cost*: nodes should be small and inexpensive; GPS integration would increase the production cost of large quantities of nodes dramatically.

2.3 Measurement principles in WSNs

To understand localisation in WSNs more thoroughly, the principles of distance measurement are discussed briefly in this section.

Some wireless localisation methods focus on iterative refinement techniques for the somewhat erroneous measurements while others assume accurate distance measurements in order to focus on other aspects of sensor node positioning; the method presented in this research being an example of the latter case.

There exist three main principles of measurement that can be used to obtain distance estimates in sensor positioning methods [13]:

- received-signal strength (RSS),
- angle-of-arrival (AOA), and
- propagation-time based systems; those include:
    - time-of-arrival (TOA),
    - roundtrip-time-of-flight (RTOF), and
    - time-difference-of-arrival (TDOA).

Figure 2.1 illustrates these.

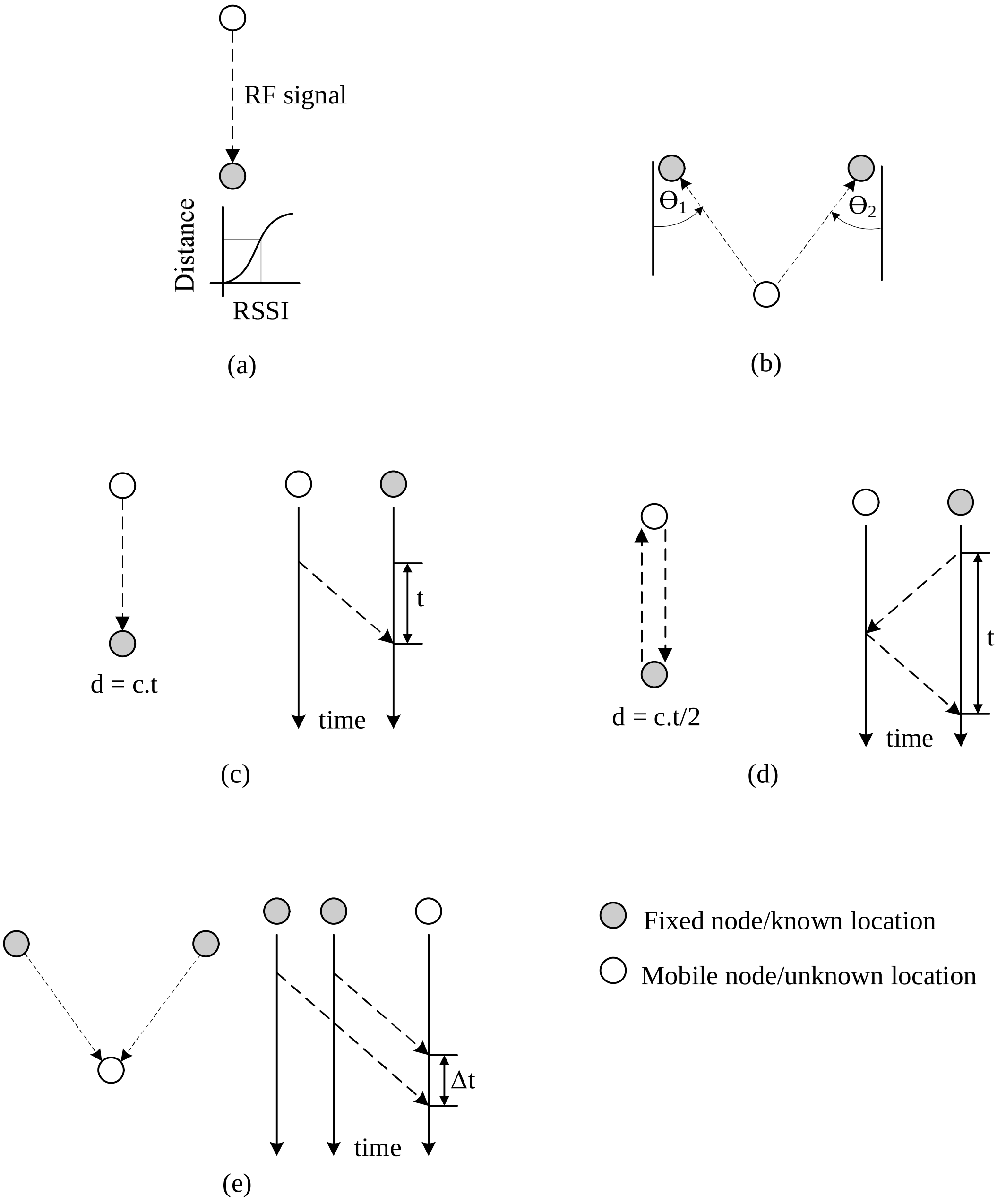


Figure 2.1 Principles of measurement in WSNs, (a) RSS, (b) AOA, (c) TOA, (d) RTOF and (e) TDOA

2.3.1 Received Signal Strength (RSS)

RSS systems measure the distances between nodes based on propagation-loss models and is the simplest/easiest method to implement, due to the fact that most RF-chips provide the ability to determine the received signal strength as a standard feature. However, indoor environments or any obstacles in sensor networks do result in a dominant multipath fading and shadowing effect.

To overcome this problem, either advanced propagation models have to be developed or the actual field distribution near the phenomena has to be measured. RSS has a nonlinear input-output mapping, thus requiring sophisticated algorithms or neural networks to be used to derive propagation-loss models.

Consequently, implementing a sensor positioning system within an RF-communication RSS based system is very much a software topic and proprietary hardware is not required.

2.3.2 Angle of Arrival (AOA)

AOA systems base position calculation on Goniometry [14]. The angle or bearing, relative to nodes located at known positions, is measured with the use of directional antennas or antenna arrays. Intersections of these directional pointers are used in calculations to obtain the position values.

The accuracy of AOA is also limited by multipath fading and shadowing effects. Another disadvantage of AOA is the requirement of specialised hardware for angle measurements.

2.3.3 Propagation-time based systems

The most accurate approach for distance measurement in sensor networks is to measure the roundtrip time-of-flight (RTOF) of the signal, travelling from the transmitter to the measuring unit (sensor node) and back [13]. Based on several such measurements, a 2D or 3D position can be derived almost directly by using a method such as lateralisation.

TOA measures one-way propagation time to calculate the distance between the measuring unit and the signal transmitter.

Both TOA and RTOF require precise time synchronization of all involved parties – fixed and mobile units. This is indeed a very difficult task to fulfil.

TDOA tries to alleviate this constraint by using the time-difference between pair-wise received signals. The benefit of TDOA systems is that they only need to synchronise the measuring units, and not the nodes in the network. This is usually achieved by using a reference transponder in a known location.

2.4 Positioning in WSNs

A beacon node is a node with a-priori knowledge of its location. This location information could either be pre-programmed on the node (the node should then be installed at that location), or any other method such as GPS can be used to obtain coordinates prior to the localisation process. A beacon node can also be called a Base Station (BS) or a landmark node.

Base stations are used to compute/estimate node coordinates (centralised manner) or to provide beacon signals with their known location information available to surrounding nodes (distributive manner). These BSs can be computationally more powerful or transmit further. Several published solutions are based solely on the availability of such powerful nodes [12], [15] – [23].

While it is possible to utilise beacon nodes in pre-planned sensor networks, it is unrealistic to assume the presence of nodes with a-priori location knowledge in rapidly deployed sensor networks (e.g. dropped from a plane).

Classification of localisation methods in WSNs can be looked at from two, almost “orthogonal”, viewpoints [3]. The first being based on the processing scope/principle embraced in the positioning algorithm:

- *centralised processing*: all nodes route their distance data to a centralised node/access point; this (usually more powerful) node will compute all the node positions in the network and forward it back to them,
- *distributed processing*: nodes are responsible for their own position estimation; they set up a local coordinate system which is then converged to form a network-wide coordinate system, and

- *localised processing*: this is distributed processing that involves only those nodes currently involved in an activity/interest.

The second classification is related to the directionality of the network coordinate systems:

- *absolute positioning*: the coordinate system of the network (and therefore the nodes) is aligned to common coordinate reference systems such as those used in the military or commercial systems (e.g. GPS); i.e. it has global coherence, and
- *relative positioning*: the coordinate system in the network is converged to a local reference system that could be arbitrary in direction, but still result in network-wide coherence.

2.4.1 Processing principles

2.4.1.1 Centralised processing

A centralized positioning algorithm enables the use of network-wide and (possibly) global information that can potentially improve the accuracy of position estimates.

Centralised processing is employed in algorithms such as the one proposed in [18], where all nodes forward their connectivity information to a central node. This node then develops a graph of the network by solving an optimization problem with the vertices and edges of beacon and non-beacon nodes as input.

The feasibility of such a system is demonstrated in [24], by generating an ad-hoc RF network that has the ability to relay connectivity statistics to a central computer.

Maximum likelihood estimators are used in a centralised manner for position calculation in [23] and [25]. Every node measures the power level of the received signals from other nodes and averages these over time to obtain pair wise power estimates – the power received at node i from node j. These are then sent periodically to a centralised computer that performs the ML-process [23].

Another ML-based centralised algorithm is proposed in [16], where they assume the presence of more than the minimum[3] number of beacon nodes in a network – 2 beacons in 2D and 3 in a 3D location. They use the redundant information present in TOA measurements to formulate several hypotheses and choose the best one, instead of just averaging the results of all possible hypotheses as in previous approaches.

Convex optimisation

A methodology for position estimation is formulated using linear and semi-definite programming based on connectivity and pair wise angles between nodes [18].

They consider connection constraints individually as the methods allow for the convex position constraints to be collected into a single optimisation problem. Figure 2.2 presents single constraints with the dark node as the landmark/beacon node and the white node whose position is constrained by sensing signals from the landmark node.

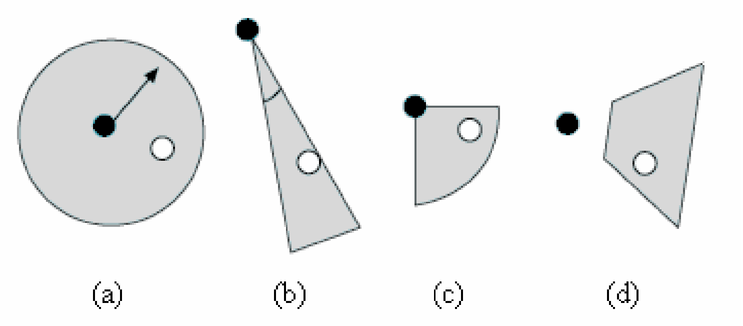


Figure 2.2 Geometrical interpretation of single constraints (a) radial, (b) angular, (c) quadrant and (d) trapezoid – adapted from [18]

Convex optimization methods use simple-to-model hardware that provides ranges or angles and simple connectivity. Efficient computational methods for most convex programming problems are also available [26]. The solutions give optimal position estimation, provided the

---

[3] A minimum number of beacons is not applicable in beacon-less networks, such as the one proposed in this research.

model can be cast as a set convex constraints. These optimal solutions are produced at the cost of centralization and the need to handle large data structures.

MDS-MAP

MDS-MAP is presented in [27]. The algorithm makes use of only connectivity to provide positions in a network with or without beacon nodes. It is based on multidimensional scaling (MDS), a method that finds an embedding in a lower-dimensional space for a set of objects, characterized by pair-wise distances between nodes.

MDS-MAP has three steps:

1. It roughly estimates the distances between each possible pair of nodes by using an all-pairs shortest-path algorithm.
2. Then it employs MDS to derive node locations that fit the estimated distances.
3. Finally, the method normalises the resulting coordinates to take any nodes whose positions are known into account.

The simulation considers a relatively uniform distribution of nodes and results suggest that the algorithm is more robust to measurement error than previous proposals. Unlike the case of convex optimization, the complexity of MDS-MAP has a theoretical bound. Furthermore, it has a wide range of applicability due to its ability to work with both simple connectivity and range measurements to provide both absolute and relative positioning (dependant on the presence or absence of beacon nodes) – again at the cost of centralisation though.

The advantage of centralised processing is that more rigorous localisation algorithms can be run to perform coordinate estimation. On the downside, this kind of processing has several drawbacks [12]:

- To forward all the necessary information to a centralised node, the route to that node must be known. This implies the use of a routing protocol other than location-based prior to location establishment and results in an unnecessary message overhead.
- It presents a time synchronisation problem. Whenever there is a change in the network topology, the central node needs to know about it. It needs to cache node locations to

ensure consistency of event reports in space and time, in order to correctly keep track of events.

- The battery lifetime of the central node will be severely affected. So will those of the nodes located close to the central node.
- Pre-planning is necessary for the location of the central node to ensure that it is easily reachable from all other nodes in the network.
- The minimum of data aggregation can be performed, since the raw data is required for location estimation by the centralised algorithm.

Overall, a central processing approach will reduce network lifetime, hamper the ease of deployment and increase the complexity of the algorithms.

2.4.1.2 Distributed processing

In distributed processing, nodes make use of several computing and communication capabilities, with the advantage of not relying on a single point of failure. It does not require a specialized/powerful central node and performs load balancing by making use of geographically distributed processors.

The difficulties of centralised processing can be alleviated if a distributed approach is followed. This will allow:

- topology changes to be updated faster with less message overhead,
- a reduction of implementation difficulty, because location-based routing can be performed from the beginning of deployment, and
- a more robust network – localisation is more tolerant to node failures because nodes are responsible for their own position estimation.

Distributed processing is utilised in a number of published localisation algorithms [12], [23], [27] – [31]. Nodes are responsible for their own location estimation. The initial coordinate establishment is sometimes followed by an iterative position refinement stage.

Collaborative multilateration is presented in [12]. It enables sensor nodes to accurately estimate their locations by using known beacon locations that are several hops away. Node locations are computed by setting up and solving a global non-linear optimization problem for two computation models, a centralized and a fully distributed approximation of the centralized model.

The results show that the fully distributed model enables the resource constrained sensor nodes to collectively solve a large non-linear optimization problem that none of the nodes can solve individually. This produces significant savings in computation and communication cost.

Figure 2.3 evaluates energy consumption of the centralised and distributed processing types in terms of the number of bytes transmitted during localisation. They consider a network of variable size and keep the density fixed at a connectivity degree of D = 6 neighbours. It is shown that the communication overhead of the centralised method is six to ten times more than that for the distributed method.

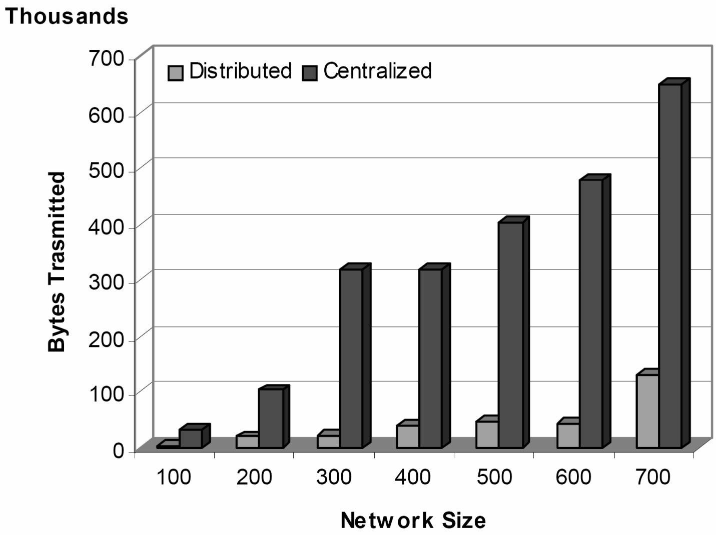


Figure 2.3 Traffic in centralised and distributed scenarios with 10% beacon nodes – adapted from [12]

The Ad Hoc Positioning System

APS (Ad-hoc Positioning System) is a hybrid between distance vector (DV) routing, and beacon based positioning (e.g. GPS) [30]. DV routing because information is forwarded in a hop-by-hop fashion, independently with respect to each landmark, and beacon based because each node estimates its own position, based on the landmark readings it gets. The APS concept has been shown to work using range [31], and angle measurements [17].

Nodes immediately adjacent to a landmark get their orientations/ranges directly from the landmark/beacon node. The rest of the nodes get their position by the use of induction: assuming that a node has some neighbours with an orientation/range for a landmark, it computes its own orientation/range with respect to that landmark and forwards the new coordinates further into the network.

APS utilises several propagation methods to compute this induction step for any combination of local capabilities: none (mere connectivity), ranging, AOA, AOA + ranging, and more.

Positive aspects of APS include the ability to support some limited mobility, accommodation of a wide degree of ranging possibilities, and distributed processing. Negatives include the requirement of a uniform distribution of landmarks and the fact that the DV nature of the algorithms will face increasing costs in the face of high node mobility [30].

The Ad-Hoc Localization System

The Ad-Hoc Localisation System (AHLoS) defines three types of multilateration:

- atomic,
- iterative, and
- collaborative.

Atomic multilateration makes up the basic case where an unknown node can estimate its location by applying basic trilateration, provided that it is within range of at least three beacon nodes (Figure 2.4a). Nodes that manage to obtain a location behave as landmarks for other nodes in an iterative manner.

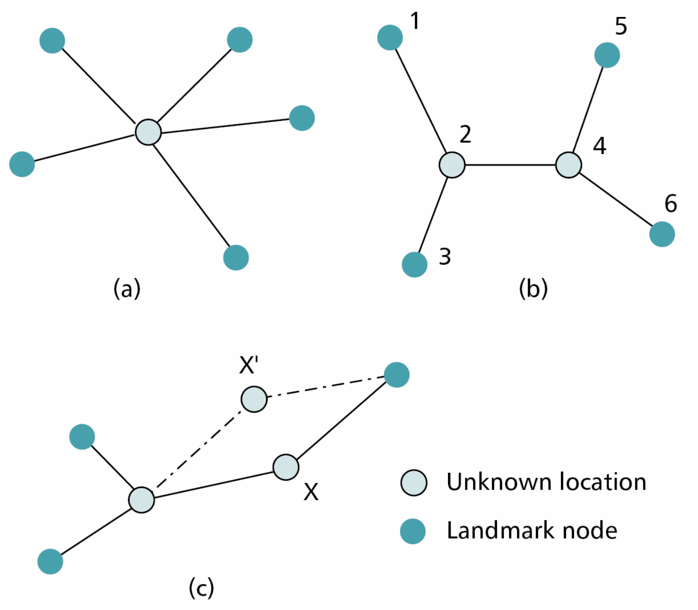


Figure 2.4 Multilateration examples – adapted from [3]

After applying both atomic and iterative multilateration, there are still nodes that are unable to find enough beacon nodes to satisfy the condition for trilateration. Figure 2.4b shows an example: neither node 2 nor node 4 may become landmarks and the problem must be considered in a collaborative fashion.

In Figure 2.4b there are five edges, thus a set of five equations can be obtained and the location is solved. In some cases, it may occur that a well determined system of *n* equations and *n* unknowns exist but will still not yield a unique answer. Figure 2.4c is such a case where node *X* can have two possible positions, *X* and *X'*, that would satisfy the system. If such a case occurs, they propose the use of optimization methods such as gradient descend [32] and simulated annealing [33] to solve the resulting system of non-linear equations.

Given a good ranging method, AHLoS is likely to produce high quality results. The disadvantage of AhLOS is that it requires a high percentage of landmarks in order to achieve a high percentage of resolved node coordinates. For example, with an average node degree of

6.28, to resolve 90 percent of the regular nodes requires a 45 percent of the nodes to be beacons [3].

2.4.1.3 Localised processing

Localised processing is a version of distributed processing that involves only those nodes currently involved in an activity/interest.

Such an algorithm is presented in [34] and is based on [35]. A *distributed leader election algorithm* selects the sensor that matches the announced task best, by minimizing a distance measure between the task and the sensors.

Sensors are sequentially selected by a task decomposition method that generates and announces the residual tasks to cover the matching error. This continues in an iterative manner until all residual tasks are covered. The sensors selected by this algorithm then form a sensor group for the originally announced task [35].

This algorithm has an inherent communication overhead due to the flooding mechanism used for task broadcasting and leader election. The overhead is reduced significantly in [34] by introducing a location aided method for task broadcasting and leader election. The sensor group, now contributor group, is dynamically maintained to confine the subsequent leader election and task announcements to this small group of sensors.

The advantage of this algorithm is that nodes, not selected for the contributor group, can be set to an energy saving state. The algorithm assumes a much synchronised network though, and therefore imposes the use of a more expensive MAC layer, due to the many synchronisation updates.

2.4.2 Relative positioning

Relative positioning provides a coordinate system that is converged to a local reference system. This reference system could be arbitrary in direction, but still results in network-wide coherence.

In most relative positioning methods it would be a trivial task to form an absolute coordinate system once the relative system is established. By substituting relative coordinates of some of the sensor nodes with known global coordinates (e.g. by means of GPS), a simple coordinate propagation procedure could be executed to update the new node locations.

The Self Positioning Algorithm

The Self Positioning Algorithm (SPA) [36] is another example of localised processing and determines positions in a coordinate system using a location reference group (LRG).

Each node in the network exchanges with its neighbours information which include the node IDs and the respective distances between itself and the other neighbours. This results in the availability of second-hop information at every node and is processed to produce a local coordinate system at the individual nodes.

Every node assumes that it is located at the origin of its own coordinate system. It then chooses an arbitrary neighbour node to represent its positive x-axis and another to exhibit a positive y component in 2D space. The position of any node that is a neighbour of these two nodes can be obtained, by using triangulation, to achieve a local coordinate system.

The LRG then initiates an iterative alignment procedure that reorients all the local coordinate systems to the coordinate system of the reference group. This provides network-wide coherence in node positions without the need for any beacon nodes, but the alignment procedure incurs a huge message overhead and the network is not very scalable.

These shortcomings of SPA are addressed in [37] by recognising that the cost of local computation is lower than that of communication in a power constrained scenario [1]. Clusters are formed throughout the network to reduce message exchange between nodes.

The cluster heads (CHs) compute position coordinates for their slave nodes relative to their localised cluster coordinate systems. Much fewer coordinate systems need to reorient during the alignment procedure and reduces communication cost considerably. The disadvantage of this method is that the alignment process is still carried out iteratively and message cost can actually be reduced even more. This is discussed in more detail in Section 4.

The Local Positioning System

The Local Positioning System (LPS) extends the ideas used by SPA, by developing a method for nodes to use some distance and direction capabilities (e.g. ranging, AOA and compasses) to establish local coordinate systems in which all immediate neighbours are placed [38].

It is then possible to register all these coordinate systems with the coordinate system of the source of the packet to achieve positioning only for the nodes participating in message forwarding. Each node that is touched by a trajectory message spends some computation to position itself in the coordinate system of the source of the packet – Figure 2.5.

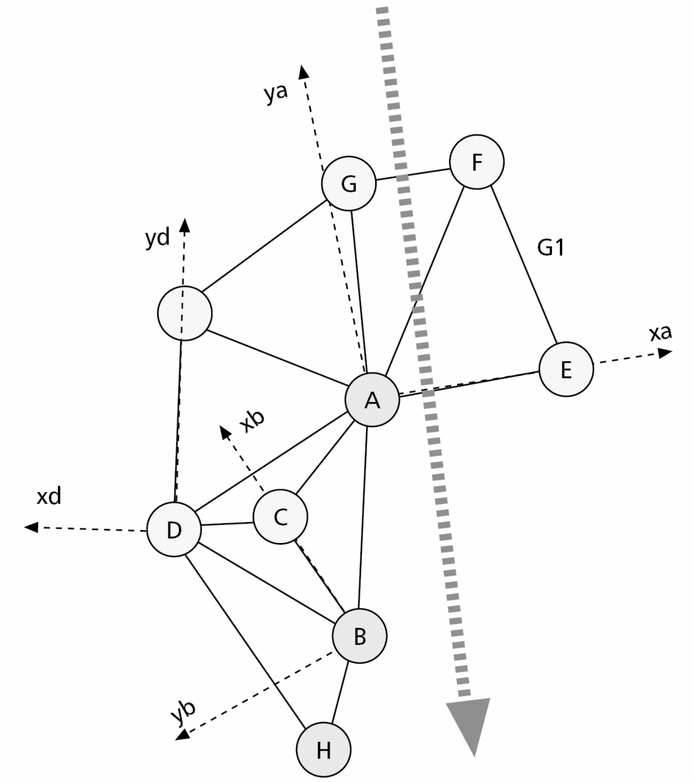


Figure 2.5 LPS positions only the nodes involved in communication – adapted from [3]

Messages are transmitted in a straight direction towards the destination node and only nodes on this line are actually positioned in the coordinate system of the source, trading off communication spent during discovery for in-node computation.

2.4.3 One-hop positioning

The main characteristic of one-hop positioning methods is that nodes are able to contact the beacons directly. The most commonly known representative of this class is the well known Global Positioning System (GPS). A disadvantage shared among most of the methods belonging to this class is the requirement for line-of-sight (LOS).

The Lighthouse project exploits the use of LOS communication in sensor node positioning [39]. The idealised principle of this method is depicted in Figure 2.6.

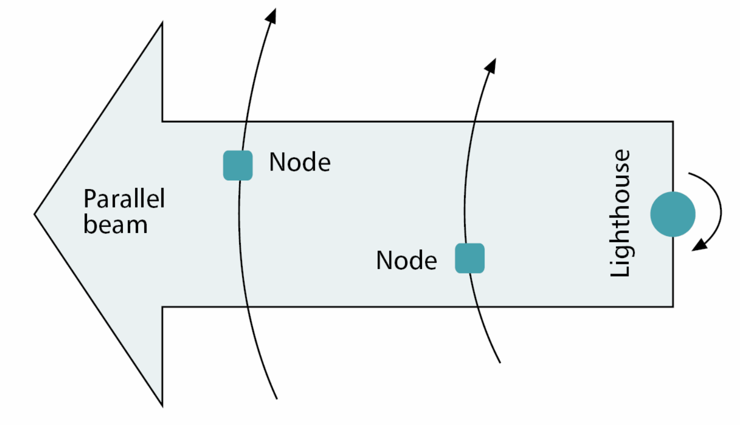


Figure 2.6 Idealistic lighthouse with a parallel beam – adapted from [3]

A parallel beam that rotates at a constant speed is used by the base station to sweep across the field of nodes. The sensor, called here *observer*, is equipped with a clock and a photo detector. The observer is able to independently estimate its range to the base station by using the known rotational speed and width of the beam, and measuring the time it sees the light.

By using three such beams it is possible to position sensor nodes, based on trilateration.

2.5 Location-based routing

Location-based routing can be regarded as a research field on its own, but because it can be deployed on top of CRCE it will be discussed briefly.

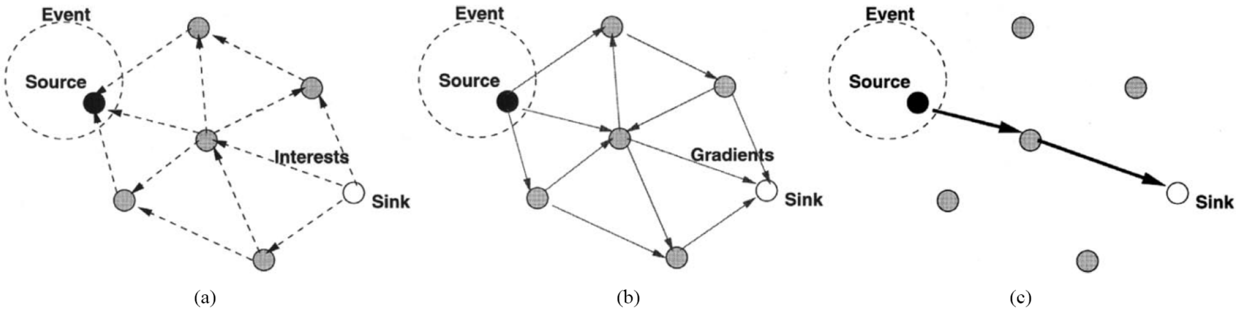
The availability of position coordinates enables scalability in most other network layer protocols, such as location-based routing. A node's address is now based on its position in the Euclidean coordinate system, with only one possible location for each node. This eliminates the problem of developing an addressing scheme large enough to accommodate millions of sensors.

It also gets rid of large routing tables. Messages can be forwarded successfully as long as the location of the message and the destination location are known. In this manner it relaxes a bit on the limited memory constraint of a tiny, inexpensive sensor node.

The simplest implementation of location-based routing is Cartesian routing [40]. Here the node assumes the next hop node in the message forwarding chain to be the one geographically closest to the destination location. Despite its perceived ease and simplicity, this method faces some logistical difficulties related to the greedy nature of the algorithm and exhibits limitations regarding energy efficient routing.

The fundamental principle of Cartesian routing is illustrated in [41], using a linear routing algorithm in a one-dimensional network topology. They then combine horizontal and vertical linear networks to form a two-dimensional network. This, now Cartesian, network is used to perform directional routing.

Attribute based routing algorithms can also use position-centric routing – providing that location is regarded as an attribute in the definition of the algorithm. A popular example of such a system is Directed Diffusion [42]



(a) (b) (c)

Figure 2.7 Simplified schematic for directed diffusion, (a) interest propagation, (b) initial gradients setup, and (c) data delivery along reinforced path – adapted from [42]

Directed diffusion consists of: interests, data messages, gradients and reinforcements. An interest message is a query or an interrogation which specifies the phenomena or data the user is interested in or concerned with – Figure 2.7a. Data is named using attribute-value pairs (e.g. location coordinates).

A sensing task is disseminated throughout the sensor network as an interest for named data and sets up gradients within the network – Figure 2.7b. A gradient is direction state created in each node that receives an interest and is set toward the neighbouring node from which the interest is received. The sensor network reinforces one or a small number of these paths while events flow toward the originators of interests along multiple gradient paths – Figure 2.7c [42].

It is worth to note that recent technological advances have lead to the emergence of distributed wireless sensor and actor networks (WSANs) [43]. This research area stems from WSNs and is capable of observing the physical world and processing the data (sensors), and making decisions based on the observations and *performing appropriate actions* (actors).

Observations made by the sensor in one location, needs to be forwarded to the actor in (possibly) another location. To perform right and timely actions, sensor data must be valid at the time of acting. Although the main concern of [43] is not localisation or routing, it is clear that location-based routing (of variant thereof) will play a major role in this new kind of network.

Refer to [44] for a survey of many more strategies and solutions related to position-centric forwarding and routing.

2.6 Chapter summary

This chapter introduced the current standards in wireless sensing and reasoned the undesirable use of GPS for obtaining position coordinates in sensor networks. Distance measurement principals were also discussed briefly.

The advantages and disadvantages of both centralised and distributed processing were given and the latter was shown to be the preferred processing method. Examples of these were also provided.

Relative positioning was discussed and examples of current methods were provided. It was stated that it would be a trivial task to form an absolute coordinate system once the relative system is established – by substituting relative coordinates of some of the sensor nodes with known global coordinates, a simple coordinate propagation procedure could be executed to update the nodes with their new locations.

It is then concluded that the desirable method for localisation would be one that is relative in positioning and incorporates distributed processing to share the processing load and lighten the communication burden.

CHAPTER 3 CLUSTER-BASED RADIAL COORDINATE ESTABLISHMENT

This chapter describes CRCE in detail with the background provided in Section 3.1. Section 3.2 discusses the network topology. A description of the MAC layer on which CRCE is based follows in Section 3.3. The message types used in CRCE are discussed in Section 3.4, and finally the CRCE algorithm is presented in Section 3.5.

3.1 CRCE background

CRCE is designed to be energy efficient. The total energy consumption (E_T) in a wireless sensor can be divided into two main parts [45]:

- E_{TX}: the energy used to transmit, receive and amplify data, and
- E_P: the energy used to process the data.

Thus, we can write:

$$E_T = E_{TX} + E_P \tag{3.1}$$

E_{TX} dominates E_P [46], therefore, by minimising the number of messages transmitted by the nodes, one can effectively reduce the energy consumption of the entire network.

Scalability is also a critical factor in WSNs. When the number of nodes in the network increases, the number of messages necessary to establish a network-wide relative coordinate system must increase linearly, as opposed to exponentially. This will enable the deployment of large networks of sensors.

CRCE is based on two main principles in order to reduce message cost. These are:

- cluster-based approach in the network topology, and
- radial coordinate update propagation in the relative convergence phase.

3.2 CRCE Network topology

The network topology of CRCE is one composed of clusters and presented in Figure 3.1. Each cluster consists of one *master* node (the cluster head (CH)) and some *slave* and *border* nodes.

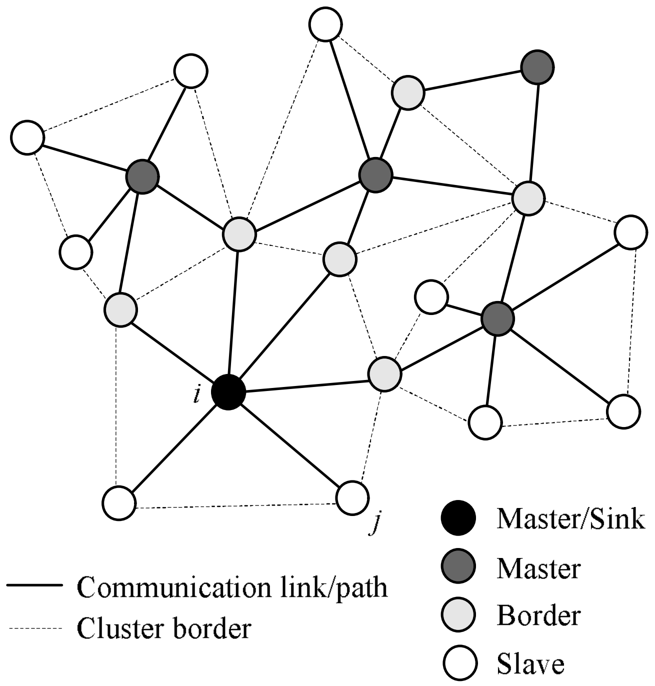


Figure 3.1 Different types of nodes

The master nodes compute the local coordinate systems of clusters and coordinate relative convergence in the network. One of the master nodes in the network will assume the role of a *sink* node. The sink initiates the coordinate establishment process once the network is deployed.

Slave nodes are one-hop neighbours of master nodes. The ability to reduce message cost depends largely on the master-slaves relationship. Slaves do not communicate directly (except in the very first stage of coordinate establishment) and provide a cluster-based environment in which message cost reduction is possible.

The term, *one-hop* neighbour, is defined. Nodes j and i are called one-hop neighbours if they can communicate with each other directly (in one hop). Moreover $\forall i \in N$, a set of nodes K_i, is defined such that $\forall j \in K_i$ and $j \neq i$, then j is a one-hop neighbour of i, where N is the set of all the nodes in the network. K_i is then the set of one-hop neighbours of i. Also, $\forall i \in N$ the set D_i is defined as the set of distances measured from node i to nodes $j \in K_i$.

The border nodes are essentially slave nodes that are part of two or more adjacent clusters. Put differently, a border node is a slave node within communication distance of more than one master node. These nodes are used for coordinate establishment purposes.

Note that the classification of nodes is only logical. Physically, these nodes are identical.

Figure 3.2 and 3.3 give simple observational comparisons between cluster-based and clusterless network topologies regarding message cost. A message is propagated from node ***a*** to node ***b***.

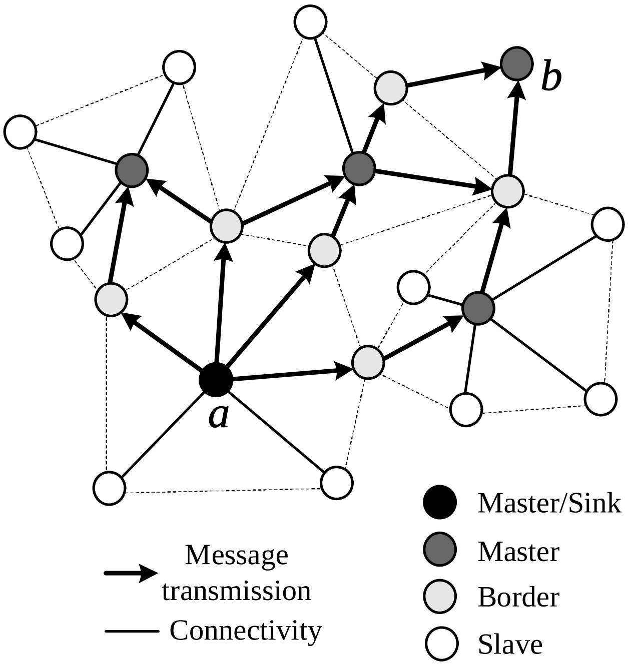


Figure 3.2 Message propagation from node ***a*** to node ***b*** in a cluster-based network

Master nodes know only the addresses of their slave/border nodes and first try to match the destination address of the message to one of them. If this is unsuccessful, the message is transmitted to all border nodes and they forward it to their other master nodes until the destination is reached. In this way communication is limited to only a subset of nodes in the network.

Figure 3.3 presents the same network of nodes, but this time with a clusterless topology.

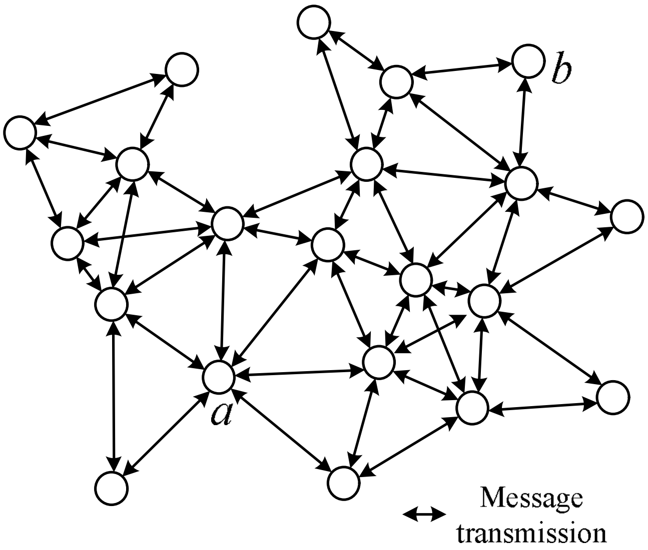


Figure 3.3 Message propagation from node *a* to node *b* in a clusterless network

Messages are forwarded to all neighbours in the clusterless setup. It is observed that clusterless topologies are much more expensive than their cluster-based counterparts.

3.3 MAC layer

Although CRCE is a network layer protocol, we need to take a brief look at the MAC/data link layer on which CRCE is based. It determines the number of messages transmitted in a broadcast or directed transmission. This information is required in the message cost simulations.

Carrier Sense Multiple Access – Minimum Preamble Sampling (CSMA-MPS) is a very low power, fully distributed MAC protocol proposed in [47]. Figure 3.4 uses the OSI model to put this into perspective.

Higher layers	Layers 4 - 7
Network	CRCE
Data link	CSMA-MPS
Physical	Sensor hardware

Figure 3.4 CSMA-MPS in the OSI model

CSMA-MPS is geared towards minimising the four main sources of energy waste identified in [48], namely:

- idle listening,
- collision of packets,
- overhearing, and
- control overhead.

Since the traffic in a wireless sensor network is expected to be low, nodes should be powered down most of the time. This introduces the problem of maintaining network connectivity without wasting too much energy during the wake-up process.

CSMA-MPS achieves this by making use of wake-up sequences that are as short as possible (by means of alternating transmit and receive slots). It also uses high bit rate transceivers that allow a lower duty cycle operation and faster transitions between different states. It does not require dedicated synchronisation messages, reducing energy consumption, and is optimized for low cost transceivers.

When a node needs to broadcast a message to a number of nodes, e.g. a master node to its slaves, CSMA-MPS does not simply transmit one message and all neighbours pick up on it (i.e. nodes are synchronised to different timeslots). It has the timing information of all its neighbours and contacts every node individually. Figure 3.5 illustrates this.

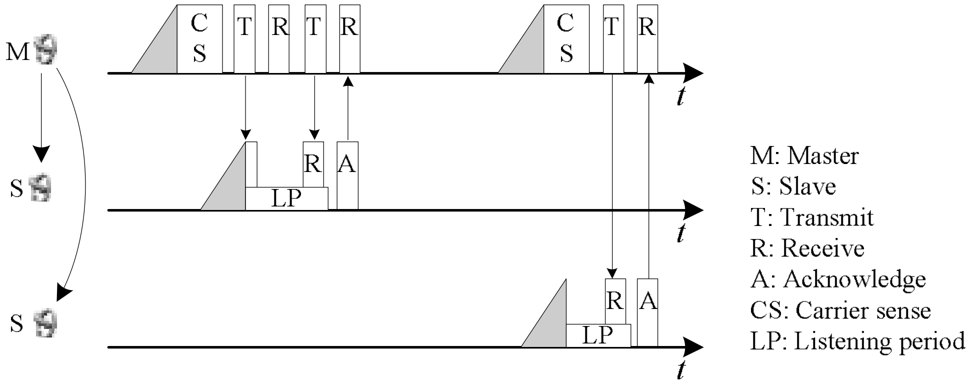


Figure 3.5 CSMA-MPS timing when the master broadcasts the same message to its slaves

The characteristics and advantages of CSMA-MPS are given:

- Nodes synchronize when they first exchange packets and in every data exchange phase by including the clock offset in the acknowledgment packet, thus periodic synchronization is not required by the nodes.
- Overhead at the transmitter is reduced due to a very short preamble length and at the receiver due to short listening times for the wake-up signal.
- At most, each node keeps synchronization information of all its direct neighbours and network-wide synchronization is not required.
- The wake-up message already contains part of or the full data portion because sensor information is typically a few bytes and much shorter than the preamble, header and CRC code together. In this way, the transmission of additional data packets with all its overhead can be avoided.
- Energy is saved by making the time of the carrier sense period very short.

CSMA-MPS is clearly a very efficient lower level protocol. The message cost simulations of CRCE is therefore based on nodes with CSMA-MPS at the MAC layer.

3.4 CRCE message types

CRCE makes use of different message types to accomplish network-wide coordinate convergence. Seeing as this is a network layer protocol, these messages are called *packets*.

Optional data from higher layers are encapsulated the by adding the necessary network layer header information to it. A typical encapsulation process is given in Figure 3.6.

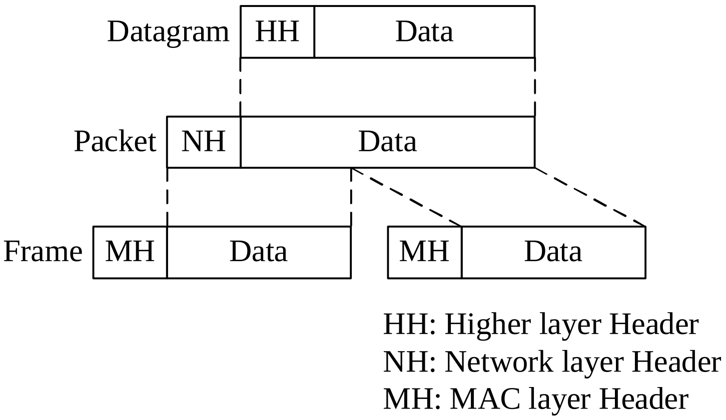


Figure 3.6 Typical encapsulation of messages

The MAC layer manages the division of packets into frames, thus we need not be concerned about it here.

A preamble precedes every CRCE packet header.

Destination address	Source address	Packet type

Figure 3.7 Header preamble structure

Destination address

This specifies the address of the destination node. A directed message will have the end node's address in this field. If a node broadcasts a message, it is proposed that this field contains a broadcast address (e.g. a string of 1's), to indicate to the lower layers that the packet should be broadcasted. The lower layers will then send this packet to all nodes within communication range.

The advantage of this is that there will be less communication between layers as opposed to handing down packets with individual node addresses (they are known) thus, saving processing energy.

Source address

This field always contains the address of the sending node.

Packet type

The type of the packet is specified in this field. Nodes use this information to distinguish between the packets for use in different stages of the CRCE process and can be implemented by using three binary digits.

There exist six types of packets. The header for *packet 1* and *2* consists only of the preamble (Figure 3.8). These packets are used for distance recording and node type establishment in the first stage of the CRCE process.

Header preamble

Figure 3.8 The basic packet header for both type 1 and 2 packets

The header for *packet 3* carries information on all the one-hop neighbours of the nodes. This includes the node ID’s, node types (master, slave or border) and distances to these nodes. These are sent by all slave and border nodes to their masters. The master nodes record this information and use it to compute local and relative coordinate systems for themselves and their slaves.

Header preamble	Neighbour address	Neighbour type	Neighbour distance
	Neighbour address	Neighbour type	Neighbour distance
	Neighbour address	Neighbour type	Neighbour distance

↓
N

N: Number of one-hop neighbours

Figure 3.9 Header structure for packet type 3.

Packet 4's header contains the coordinates of the slaves. Each slave receives such a packet from its master which tells it what its position is. The slave then saves this information and sets its coordinates accordingly. It also stores the ID of the node to which its master converged to.

Header preamble	Converged ID	Coordinate

Figure 3.10 Header structure for packet type 4

The header for *packet 5* carries the correction angle (ω_{ik} – see Section 3.5.2.2) together with the same information as in packet 4's header. This packet is sent by the masters to their border nodes.

Header preamble	Converged ID	Coordinate	Correction angle (ω_{ik})

Figure 3.11 Header structure for packet type 5

Packet 6's header contains the correction angle present in packet 5 and is forwarded by border nodes to all master nodes connected to the border node, except for the master from which it received its already converged coordinates. This angle is used by the receiving master node to converge to the network coordinate system. It also contains the converged ID – the ID of the node to which system it reorients to.

Header preamble	Converged ID	Correction angle (ω_{ik})

Figure 3.12 Header structure for packet type 6

3.5 CRCE algorithm

As already stated, the CRCE method makes use of a cluster-based network with a radial convergence process. After network deployment, a designated sink node starts the CRCE process. Sink selection in the network can be achieved in one of two ways:

- pre-programming; prior to deployment, one node can be pre-programmed to be the sink node, or
- by executing a sink selection algorithm in the network prior to the CRCE process.

The latter of the two is not the preferred method for sink selection, as this results in a message overhead and will therefore not be energy efficient. Sink node pre-programming does not involve any kind of positioning or coordinate specification and will therefore not affect the ease of node deployment.

CRCE consists of two stages, each comprising of two phases:

1. Neighbour discovery stage
    a. Distance and node type establishment phase
    b. Neighbour distance reporting phase
2. Coordinate establishment stage
    a. Local coordinate calculation phase
    b. Relative coordinate convergence phase

These are described next.

3.5.1 Stage 1: Neighbour discovery

The objective of this stage is to enable a master node to obtain necessary information on all its one and two-hop neighbours. At the end of this stage the master node will have recorded all distances to its one-hop nodes, as well as the distances between its one and two-hop nodes.

This will enable coordinate establishment in *Stage 2*. The algorithm for the first stage of CRCE is given (Algorithm 3.1).

```
if (sink)
  broadcast(P1)
end if
if (unknown_node)
  wait (P1) OR (P2) and record distance
  if (P2 received)
    init(random_timer)
    decrement(random_timer)
    if (timer = 0) AND (no P1 received)
      status = master
      broadcast(P1)
      wait (P2s) and record distances
    else if (P1 received)
      record(distance)
      if (more than one P1 received)
        status = border
      else status = slave
      broadcast (P2)
    end if
  else if (P1 received)
    record(distance)
    if (more than one P1 received)
      status = border
    else status = slave
    broadcast (P2)
  end if
if (slave) OR (border)
  send (P3) to master(s)
end if
```

Algorithm 3.1 CRCE algorithm: Stage 1

3.5.1.1 Phase 1: Distance and node type establishment

At the beginning of Stage 1, no node, except the sink, knows its type (master, slave, or border node) and no messages have been exchanged between nodes.

The sink node starts the process by broadcasting P1 (packet 1) type messages to all surrounding nodes. Every node receiving a message, record the distance to the master (in this case the sink) and label itself as a *slave* node.

These nodes, in turn, broadcast P2 messages which will trigger the countdown of a random timer at the nodes that are not yet classified. All nodes that receive P2 messages record the distances between themselves and the sender.

If the timer of a node expires before it receives a message from any other master node (P1 type message), it broadcasts a P1 message that stops the countdown at any node in its range/domain, ultimately establishing itself as a master and the other as slaves. Nodes that are shared by two or more masters will then classify themselves as border nodes. All nodes receiving this P1 message record their distances to the master and only those that have not previously broadcasted P2 messages do so now.

The cycle repeats until all nodes are classified and all distances between them are recorded.

To give an example of node type establishment, Figure 3.13a depicts a network of 9 nodes together with a randomly chosen timer value for each. According to the method described here for neighbour discovery, the node with an initial timer of 0.1 will establish itself as a master node. It broadcasts this to its neighbours. They stop decrementing their timers and label themselves as slave nodes.

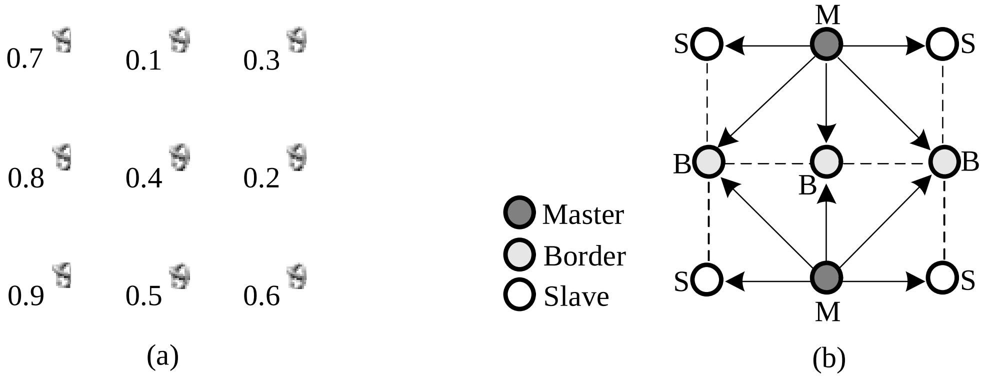


Figure 3.13 Node type establishment in CRCE

The next node's timer to reach zero is then the one with a timer of 0.5. It does the same as the first one and so do its slaves. The slaves with more than one master node classify themselves as border nodes. Figure 3.13b illustrates the network setup at the end of Stage 1.

3.5.1.2 Phase 2: Neighbour distance reporting

Every node now has a set of known distances to all its neighbouring nodes. The master nodes need to know the distances between its one and two-hop neighbours for coordinate calculation purposes. Once a slave/border node has received all distance updates from its neighbours (in the previous phase) it sends this information to its master node(s) using P3 type messages.

An example of such information available at master node 1 in Figure 3.14 is given in Table 3.1.

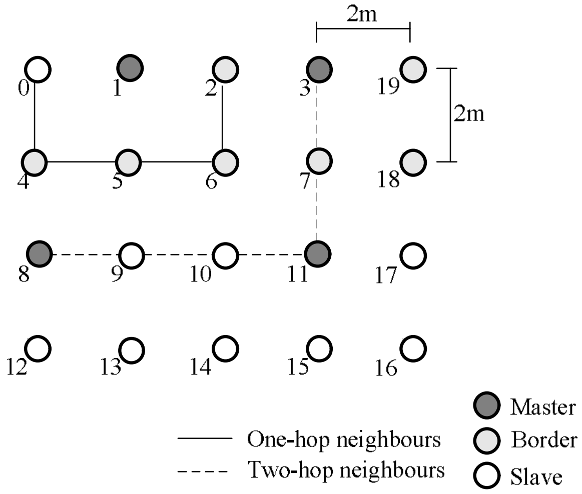


Figure 3.14 One and two-hop neighbours of Node 1

Table 3.1 Distance Information of 1st and 2nd Hop Nodes at Node 1

1st hop	Type	Distance (m) – Node 1 to 1st hop	2nd hop	Type	Distance (m) – 1st hop to 2nd hop
0	S	2			
			1	M	2
			5	B	√8
			4	B	2
2	B	2			
			1	M	2
			3	M	2
			5	B	√8
			6	B	2
			7	B	√8
4	B	√8			
			0	S	2
			1	M	√8
			5	B	2
			8	M	2
			9	S	√8
5	B	2			
			0	S	√8
			1	M	2
			2	B	√8
			4	B	2
			6	B	2
			8	M	√8
			9	S	2
			10	S	√8
6	B	√8			
			1	M	√8
			2	B	2
			3	M	√8
			5	B	2
			7	B	2
			9	S	√8
			10	S	2
			11	M	√8

3.5.2 Stage 2: Coordinate establishment

Every node in the network will have position coordinates relative to the sink node at the end of this stage. Algorithm 3.2 describes the workings of the second stage of CRCE.

```
if (master) OR (sink)
  wait (P3s) and establish local coordinate system
end if
if (sink) AND (coordinate system established)
  send(P4) to slaves
  for (all border nodes that are neighbours
       and have the same second or third master)
    send(P5) to borders
  end for
end if
if (slave) AND (P4 received)
  update position coordinates
end if
if (border) AND (P5 received)
  update position coordinates
  send(P6) to unconverged masters
end if
if (master) and (P6 received)
  translate coordinate system
  send(P4) to slaves
  for (all border nodes that are neighbours
       and have the same second or third master)
    send(P5) to borders
  end for
end if
```

Algorithm 3.2 CRCE algorithm: Stage 2

3.5.2.1 Phase 1: Local coordinate calculation

No messages are sent in this phase. Every master node calculates its local coordinate system by using triangulation, based on the distances obtained in the previous phase. It does not transmit the calculated positions of the slave/border nodes to them until *Phase2*.

The procedure for local coordinate setup in clusterless networks is presented in [36]. Minor changes needs to be made for use in the cluster-based CRCE method. The procedure is detailed next.

Every master node assumes that it is located at the origin of the network and therefore sets its own coordinate to (0, 0). Figure 3.14 represents the setup.

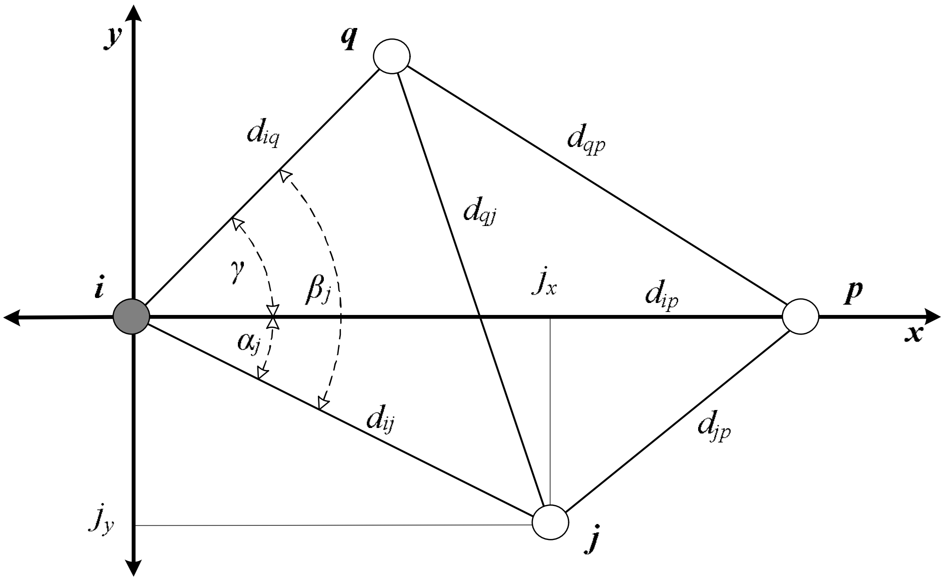


Figure 3.14 Coordinate system setup for node *i* and position calculation for node *j*

Master node *i* chooses two neighbouring nodes (borders and/or slaves), $p, q \in K_i$ such that the distance between *p* and *q* (d_{pq}) is known and larger than zero, and K_i is the set of one-hop neighbours defined in Section 3.2.

Nodes *i*, *p* and *q* cannot lie on the same line. This is because triangulation is used for coordinate computation and this would result in a line instead of a triangle. Equation 3.2 has to hold for this to be true.

$$-1 < \frac{d_{ip}{}^{2} + d_{iq}{}^{2} - d_{pq}{}^{2}}{2d_{ip}d_{iq}} < 1 \tag{3.2}$$

Node *i* defines its own local coordinate system such that node *p* lies on the positive *x* axis of the coordinate system and node *q* has a positive component q_y. The local coordinate system of *i* is now uniquely defined in terms of *i*, *p* and *q*, and their coordinates are:

$$\begin{aligned} i_x &= 0; & i_y &= 0; \\ p_x &= d_{ip}; & p_y &= 0; \\ q_x &= d_{iq}\cos\gamma; & q_y &= d_{iq}\sin\gamma. \end{aligned} \tag{3.3}$$

γ is the angle ∟(*p*, *i*, *q*) and is obtained by using the cosines rule for triangles; in triangle Δ(*p*, *i*, *q*). Equation 3.4 applies.

$$\gamma = \arccos\frac{d_{ip}{}^{2} + d_{iq}{}^{2} - d_{pq}{}^{2}}{2d_{ip}d_{iq}} \tag{3.4}$$

The positions for the nodes $j \in K_i$, and $j \neq p, q$, for which the distances d_{ij}, d_{qj} and d_{pj} are known, are computed by triangulation.

$$j_x = d_{ij}\cos\alpha_j \qquad \begin{aligned} &\mathit{if} && \beta_j = |\alpha_j - \gamma| \\ &\mathit{then} && j_y = d_{ij}\sin\alpha_j \\ &\mathit{else} && j_y = -d_{ij}\sin\alpha_j \end{aligned} \tag{3.5}$$

α_j is the angle ∟(*p*, *i*, *j*), β_j is the angle ∟(*j*, *i*, *q*) and γ is the angle ∟(*p*, *i*, *q*). Equations 3.6 and 3.7 are used to compute the values of α_j and β_j.

$$a_j = \arccos \frac{d_{ij}^{\;2} + d_{ip}^{\;2} - d_{pj}^{\;2}}{2 d_{ij} d_{ip}} \tag{3.6}$$

$$b_j = \arccos \frac{d_{iq}^{\;2} + d_{ij}^{\;2} - d_{qj}^{\;2}}{2 d_{iq} d_{ij}} \tag{3.7}$$

Given the above, the positions of the one-hop neighbours $k \in K_i$, $j \neq p, q$, which are not neighbours of *p* or *q*, can be obtained by using the position of node *i* in conjunction with two other nodes, in the domain of node *i*, for which the positions have already been calculated in this way, if the distances from node *k* to these nodes are known.

3.5.2.2 Phase 2: Relative coordinate convergence

All master nodes need to reorient their local coordinate systems only *once* to converge to that of the sink node. Coordinate updates traverse the network radially. The concept of radial propagation is described in Section 4.2.2.

The master nodes calculated the positions of their slaves in the previous phase. Using P4 type messages, these coordinates are sent to the slaves to update their positions accordingly.

The master node also sends the coordinates of its border nodes to it, together with the angle ω_{ik} (described later), destined for the border node's other master(s). This is accomplished using P5 type messages.

In turn, each border node then transmits P6 type messages to all its masters that have not yet converged to the coordinate system of the sink node. These messages contain information (e.g. angle ω_{ik}) to perform the necessary rotational and translational calculations.

The converged masters then broadcast P4 and P5 messages in their domains and the cycle continues until all nodes in the network have known positions, relative to that of the sink node.

The network is now converged.

The mechanism for determining the coordinate translation parameters for a system of three nodes which are aware of their mutual distances [36], cannot be applied directly here. The reason for this is because, by definition, two master nodes cannot be within direct range of each other and therefore do not know the distance between them. Thus, in addition to node *j*, another border node, say *l*, is required which is shared by master nodes *i* and *k* in order to obtain the angles and distances necessary for computing the translation parameters.

The mathematical formulation of the convergence process is proposed in [37] and will now be described.

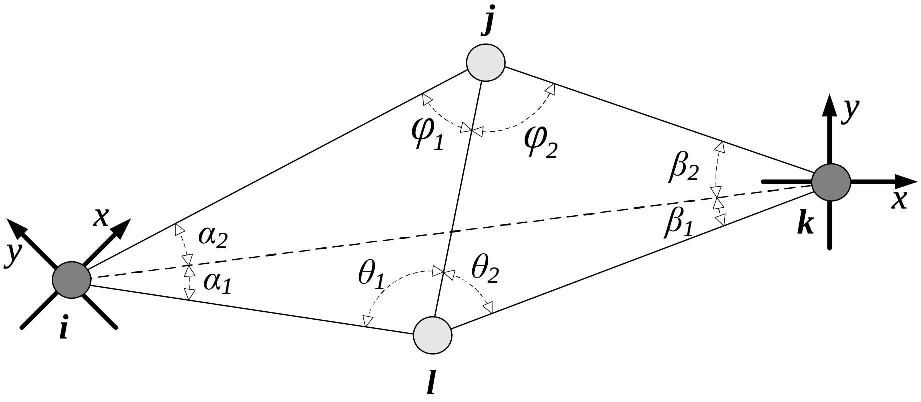


Figure 3.15 Distance calculation through triangulation

In Figure 3.15 two master nodes *i* and *k*, with common border nodes *j* and *l*, are considered. Master node *i* is assumed to have the sink property and therefore node *k* changes over to the system of node *i*. To obtain the distance d_{ik}, triangulation is used as described below.

First the angles θ_1, θ_2, φ_1 and φ_2 are obtained by using triangulation in the triangles $\Delta(i, j, l)$ and $\Delta(k, j, l)$.

$$q_1 = \arccos \frac{d_{il}^2 + d_{jl}^2 - d_{ij}^2}{2 d_{il} d_{jl}} \tag{3.8}$$

$$\theta_2 = \arccos\frac{d_{kl}^{\ 2} + d_{jl}^{\ 2} - d_{kj}^{\ 2}}{2d_{kl}d_{jl}} \tag{3.9}$$

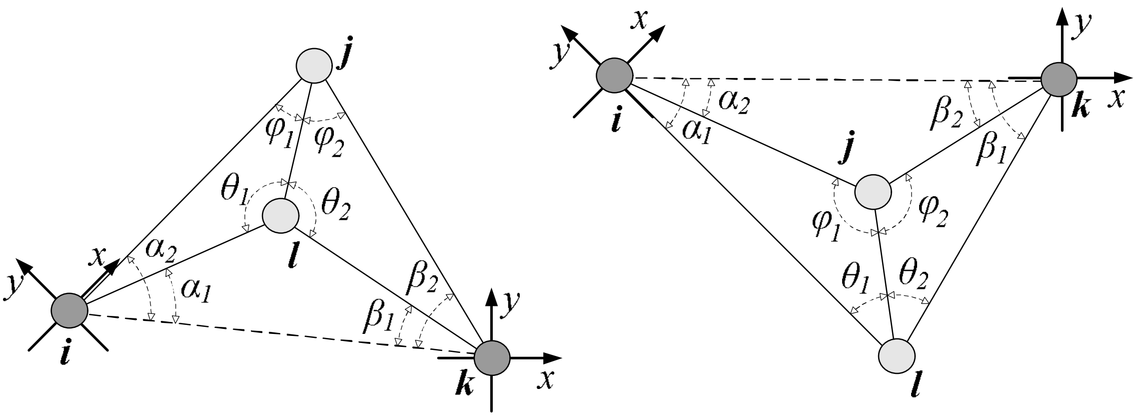
$$\varphi_1 = \arccos\frac{d_{ij}^{\ 2} + d_{jl}^{\ 2} - d_{il}^{\ 2}}{2d_{ij}d_{jl}} \tag{3.10}$$

$$\varphi_2 = \arccos\frac{d_{kj}^{\ 2} + d_{jl}^{\ 2} - d_{kl}^{\ 2}}{2d_{kj}d_{jl}} \tag{3.11}$$

Using $\theta = \theta_1 + \theta_2$, d_{ik} is calculated as

$$d_{ik}^{\ 2} = d_{il}^{\ 2} + d_{lk}^{\ 2} - 2d_{il}d_{lk}\cos\theta\ . \tag{3.12}$$

However, in the special case of Figure 3.16a, which occurs when $\theta_1 + \theta_2 > \pi$, $\theta = 2\pi - \theta_1 - \theta_2$.



(a) (b)

Figure 3.16 Special cases of quadrilateral formation

α₁ and *α₂* now need to be calculated in order to find the orientation of node *k* with respect to node *i*.

$$\alpha_1 = \arccos\frac{d_{il}^{\ 2} + d_{ik}^{\ 2} - d_{kl}^{\ 2}}{2d_{il}d_{ik}} \tag{3.13}$$

$$\alpha_2 = \arccos\frac{d_{ij}^{\ 2} + d_{ik}^{\ 2} - d_{kj}^{\ 2}}{2d_{ij}d_{ik}} \tag{3.14}$$

The final calculations for the rotation and mirroring of the coordinate system at node *k* depends on the angle ω_{ik} of the vector $\overrightarrow{ik}$ at node *i* (Figure 3.17).

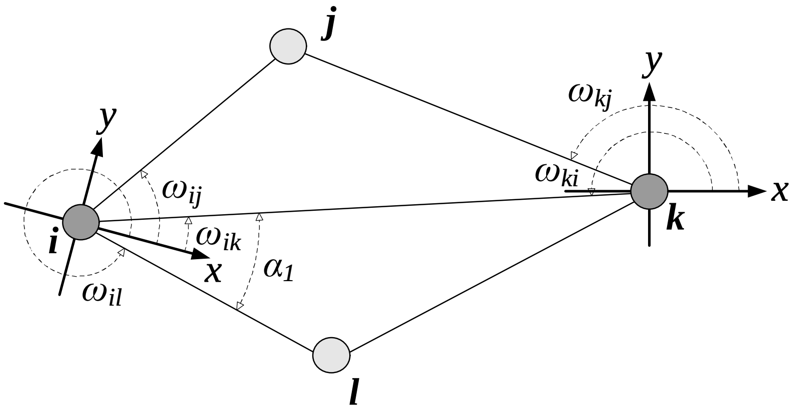


Figure 3.17 Angles of the nodes *j*, *k* and *l* from node *i*, and angles of *j* and *i* from node *k*

ω_{ik} can be obtained by subtracting or adding α_1 from ω_{il} depending on the orientation of the quadrilateral *ijkl*. If node *l* forms the lower two sides of the quadrilateral (Figure 3.15 and Figure 3.16b), α_1 is added while it is subtracted for Figure 3.16a. The node that forms the lower two sides is determined by the following:

$$\text{Lower node} = \begin{cases} l & \text{if } \omega_{il} < \omega_{ij} \text{ AND } \omega_{ij} - \omega_{il} < \pi \\ l & \text{if } \omega_{il} > \omega_{ij} \text{ AND } \omega_{il} - \omega_{ij} > \pi \\ j & \text{if } \omega_{ij} < \omega_{il} \text{ AND } \omega_{il} - \omega_{ij} < \pi \\ j & \text{if } \omega_{ij} < \omega_{il} \text{ AND } \omega_{ij} - \omega_{il} > \pi \end{cases} \tag{3.15}$$

Note that the quadrilateral in Figure 3.16b is characterised by the fact that $j > p$. Given that node *l* is the lower node, then ω_{ik} is determined by the following:

$$\omega_{ik} = \begin{cases} (\omega_{il} + \alpha_1)\text{mod}(2\pi) & \text{if } \varphi_2 < \pi/2 \\ (\omega_{il} - \alpha_1)\text{mod}(2\pi) & \text{otherwise} \end{cases} \tag{3.16}$$

If node *j* is the lower node, then ω_{ik} is determined by the following:

$$\omega_{ik} = \begin{cases} (\omega_{ij} + \alpha_1)\text{mod}(2\pi) & \text{if } \theta_2 < \pi/2 \\ (\omega_{ij} - \alpha_1)\text{mod}(2\pi) & \text{otherwise} \end{cases} \tag{3.17}$$

The same procedure can now be followed to determine ω_{ki}. Angle correction and/or mirroring of node *k*'s coordinate system is now necessary. To orient node *k* to node *i*'s coordinate system, [37] gives the following:

if $(\omega_{ij} - \omega_{ik} < \pi$ AND $\omega_{kj} - \omega_{ki} < \pi)$ OR $(\omega_{ij} - \omega_{ik} > \pi$ AND $\omega_{kj} - \omega_{ki} > \pi)$
then mirroring is necessary
the correction angle is $\omega_{ki} + \omega_{ik}$

if $(\omega_{ij} - \omega_{ik} < \pi$ AND $\omega_{kj} - \omega_{ki} > \pi)$ OR $(\omega_{ij} - \omega_{ik} > \pi$ AND $\omega_{kj} - \omega_{ki} < \pi)$
then mirroring is not necessary
the correction angle is $\omega_{ki} - \omega_{ik} + \pi$

The coordinate systems of master nodes *i* and *k* now have the same direction. Also, the direction and magnitude of vector $\overrightarrow{ik}$ is known. This information can now be used to calculate the coordinates of any node *m* in the domain of master *k* as

$$\overrightarrow{im} = \overrightarrow{ik} + \overrightarrow{km}. \tag{3.18}$$

3.6 Chapter summary

CRCE was introduced in this chapter. To reduce the energy consumption of the nodes during the localisation process, it focuses on the reduction of the number of messages necessary to establish a network wide coordinate system. This would be achieved by employing a cluster-based approach in the network topology and by letting the coordinate updates traverse the network radially.

The preferred MAC layer for CRCE was discussed briefly. The procedure for computing a relative coordinate system in a distributive manner was given together with the CRCE algorithm.

Table 3.2 summarises the packet types of CRCE, and Table 3.3 gives a summary of the different stages and phases of the CRCE process.

Table 3.2 Packet types

Packet type	Binary	Sent from	Sent to	Purpose
Unused	000_2			
1	001_2	Masters	All neighbour nodes	Distance recording and establishment of node types: itself as master and the neighbours as slaves.
2	010_2	Slaves	All neighbour nodes	Distance recording and master node discovering. Neighbouring nodes get to be masters by setting a random timer upon receipt of this packet; if the timer expires before a master node contacts it, it assumes the master node role.
3	011_2	Slave and border nodes	Master	Reports first hop neighbours' IDs along with the distances to them.
4	100_2	Master	Slaves	Updates position coordinates.
5	101_2	Master	Borders	Updates position coordinates and forwards the correction angle.
6	110_2	Borders	Masters	Forwards the information required for the coordinate translation/convergence process (correction angle).
Unused	111_2			

Table 3.3 Stages of the CRCE process

	Name	Purpose	Message transmission?
Stage 1	Neighbour discovery	Gather the distance information required for Stage 2	
• Phase 1	Distance and node type establishment	Record all distances to one-hop neighbours and classify nodes as either Master, Slave or Border along the way	Yes
• Phase 2	Neighbour distance reporting	Report all recorded distances to all one-hop neighbours in order for nodes to have distance information of their two-hop neighbours	Yes
Stage 2	Coordinate establishment	Use the information obtained from Stage 1 to setup a network-wide coordinate system	
• Phase 1	Local coordinate calculation	Every Master node calculates its own coordinate system by using the method proposed in [35].	No

• Phase 2	Relative coordinate convergence	All Master nodes now converge to the coordinate system of the sink node, using the method proposed in [36].	Yes

CHAPTER 4 COMPARISON NODE POSITIONING METHODS

CRCE is compared to three other node positioning methods. Two of them are presented in the literature and the other is a modified version of one of the two literature methods. They are attributively named as follow:

- Radial- and Clusterless [36],
- Clusterless (modified Radial- and Clusterless), and
- Radial-less [37].

The reason for choosing these methods is that they all share some common characteristics with CRCE. This makes it possible to compare them against each other and was identified during the literature study process. These are listed:

- distributed position computation – there is no central node or computing device that is responsible for position computing of the entire network,
- infrastructure-free deployment – none of the nodes are required to be pre-installed or located at specific coordinates in the network prior to network deployment,
- GPS-less or beacon-less architecture – nodes equipped with a positioning device (such as GPS) or nodes pre-programmed with position coordinates are not necessary, and
- relative coordinate system – the final coordinate system is relative to either the sink node (radial methods) or the node with the lowest ID (iterative methods).

4.1 Radial- and Clusterless

This method is proposed in [36] and is called the Self Positioning Algorithm (SPA). Similar to all methods presented here, they propose a distributed infrastructure-free positioning algorithm that does not rely on GPS. It does not have a cluster-based topology, though, and coordinate translation updates traverse the network iteratively rather than radially. We discuss this positioning method by looking at each of the two stages.

4.1.1 Stage 1: Neighbour discovery

Every node performs the following procedure:

- Phase 1
  - detects one-hop neighbours,
  - logs distances to one-hop neighbours, and
- Phase 2
  - sends this information to all one-hop neighbours.

Every node sends a broadcast message in the beginning of the network setup in Phase 1. These messages are used to log all the one-hop neighbours and to record the distances between them in the same way as in CRCE. Once this information has been obtained, it is sent to all the one-hop neighbours in Phase 2.

Each node now knows the distances to its one-hop neighbours. Also, it knows its two-hop neighbours as well as the distances between its one-hop and two-hop neighbours. This information is now used in Stage 2 to establish a relative coordinate system.

4.1.2 Stage 2: Coordinate establishment

As in CRCE, Stage 2 is again divided into two phases:

- local coordinate calculation, and
- relative coordinate convergence.

Local coordinate calculation is the same as for CRCE with the only difference that it is performed in a clusterless network. Once every node has completed building its own coordinate system in the before mentioned manner, Phase 1 of Stage 2 is finished.

We now have a local coordinate system at each node with positions (0, 0) and different coordinate system directions. Figure 4.2 illustrates this.

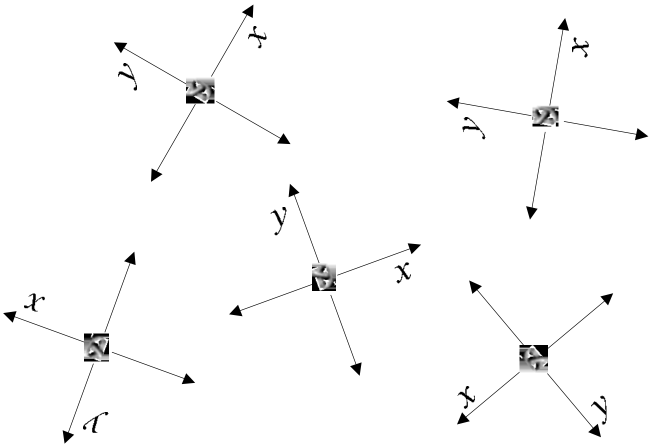


Figure 4.1 Example coordinate system directions after local coordinate computation in the RC-less method

In Phase 2, relative coordinate convergence, we need to reorient the coordinate system of every node to the direction of a network-wide coordinate system. The mathematics of how to converge each node to a network-wide coordinate system is presented in [36]. But, because this is proposed for a clusterless method, it is not applicable to CRCE and will not be described.

What is of interest though, is the way in which the coordinate updates propagate throughout the network. This accounts for the number of messages transmitted in Stage 2 of the coordinate establishment phase and has a direct influence on the energy level of the network.

The aim is for each node to reorient its coordinate system to the node in the network with the lowest ID. Every node executes the following procedure:

```
broadcast own coordinate system info
  while (not converged to lowest ID in network)
    wait(coordinate broadcast messages)
    if (received_message->converged_ID < my_converged_ID)
      converge to that node's system
    end if
    if (converged in this cycle)
      broadcast(coordinate_system_info)
    end if
  end while
```

Algorithm 4.1 Coordinate convergence of the RC-less method

From Algorithm 4.1, it is evident that coordinate convergence has an inherent iterative element in the RC-less method – the *while* loop. Figure 4.3 illustrates this even further with an example network of 16 nodes.

Every node stores the address of the node to which coordinate system it converged to. We call this the *converged ID*. The *converged ID* is sometimes the same as the *node ID*. After the local coordinate systems have been set up in phase 1 of this stage, the node ID and the converged ID of every node will match, as each node is converged to its own coordinate system.

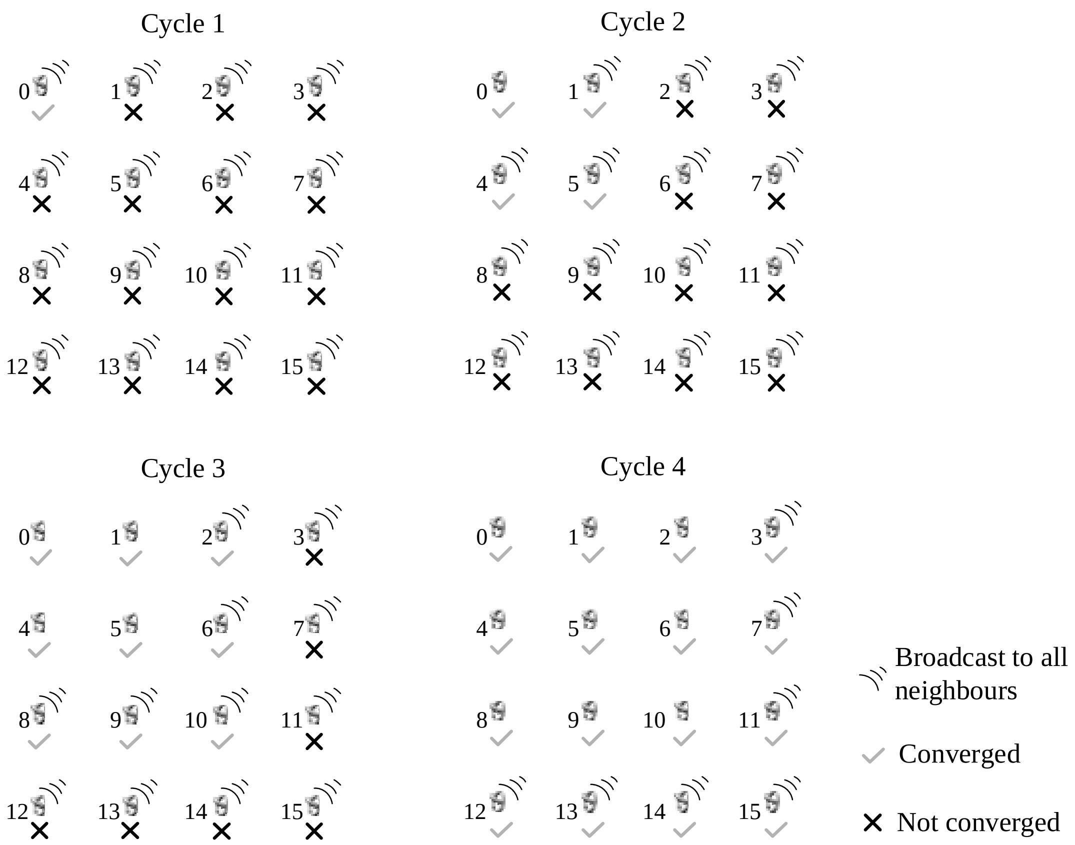


Figure 4.2 Coordinate convergence of the RC-less method

We observe 4 cycles (looped 4 times) in Figure 4.2. At the end of Cycle 1, every node has broadcasted its coordinate information to all neighbours. Node 0 keeps its coordinate orientation (it has the lowest ID) and all other nodes translate their coordinate system to that neighbour with the lowest *converged ID*. Node 0 is now the reference for the network coordinate system. It does not need to retransmit in Cycle 2 because it did not receive any coordinate translation messages from any node with a lower converged ID. All the other nodes broadcast their new coordinate information in Cycle 2. Nodes 1, 4 and 5 are now converged towards node 0's coordinate system (converged ID's = 0). They do not need to transmit in Cycle 3 because they did not receive a message from any node with a lower converged ID. This repeats until the end of Cycle 4 and illustrates the iterative process for this kind of coordinate convergence.

4.2 Clusterless

This is also a non-cluster based node positioning method. It is not found in the literature anywhere and was designed with the purpose of being a comparison method for this research to study the effect of radial convergence in clusterless network topologies. Basically, C-less is RC-less, modified to incorporate radial convergence.

4.2.1 Stage 1: Neighbour discovery

This stage is exactly the same as for the RC-less method described in Section 4.1 – both are clusterless. Neighbour nodes are discovered and distances between them are recorded. These are then sent to all one-hop neighbours for use in Stage 2, the coordinate establishment stage.

4.2.2 Stage 2: Coordinate establishment

The coordinate convergence calculations in Phase 2 are also the same as those for the RC-less method. The difference, however, is in the way these coordinate update messages are propagated throughout the network. In contrast to the iterative convergence of the RC-less method, described in Section 4.1.2, coordinate updates in the C-less method traverse the network radially.

Another difference is that the network converges to a designated *sink node*, rather than the node with the lowest ID – the same as CRCE. The sink node could be the node with the lowest ID (e.g. node 0) if it is so desired.

When the local coordinate system is already established at nodes during Phase 1, every node proceeds to execute Algorithm 4.2.

```
if(node_type = sink)
  broadcast(coordinate_update)
end if
wait(coordinate_update)
if(coordinate_update received)
  converge coordinate system
  broadcast(coordinate_update)
end if
```

Algorithm 4.2 Coordinate convergence of the C-less method

We observe that there is no loop contained in Algorithm 4.2 – radial convergence. This reduces the number of messages in the network dramatically, saving valuable energy. It is illustrated even further.

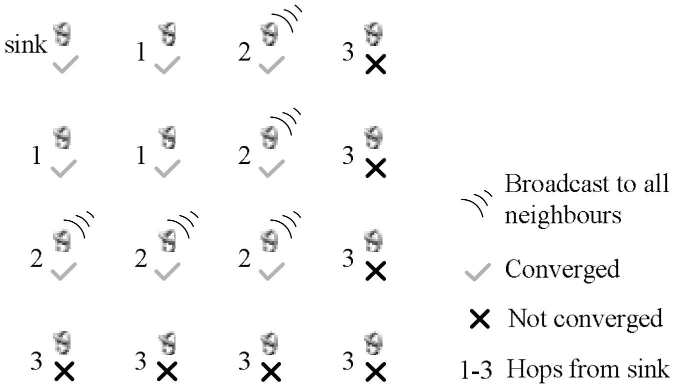


Figure 4.3 Coordinate convergence of the C-less method

Figure 4.3 is paused at the second hop in the coordinate update process. The sink node is converged to its own coordinate system at the beginning of Phase 2 of the coordinate establishment stage. It forwarded the coordinate update information to all of its one-hop neighbours.

They then reoriented their system towards that of the sink node. In turn they sent this information to all of their one-hop neighbours. These nodes (two-hop nodes in Figure 4.3) will now converge to the coordinate system of the sink and send it along to their one-hop neighbours – third-hop nodes of the coordinate update process.

This illustrates the radial convergence of the C-less method. Coordinate update propagation in the CRCE follows the same principle, the only difference being the cluster-based network.

4.3 Radial-less

This node positioning method has a cluster-based network topology and is proposed in [37]. Local coordinate establishment is performed at only a subset of network nodes (the cluster heads) and is in principle the same as for the RC-less method.

4.3.1 Stage 1: Neighbour discovery

The neighbour discovery stage of the R-less method is exactly the same as in CRCE. Every node starts to decrement a random timer once the nodes are deployed. When the timer has expired and it has not been contacted by a master node, it sends a message to all its neighbouring nodes, establishing itself as a *master*. All nodes that hear this message stop with the countdown, and classify themselves as *slaves*. If a slave receives a message from two or more master nodes, it becomes a *border* node.

In Phase 2, the slaves and border nodes send the distance information to their master nodes only. This reduces the number of messages transmitted in Stage 1 when compared to the clusterless methods.

4.3.2 Stage 2: Coordinate convergence

All the information necessary to set up a local coordinate system is now available. Instead of computing a coordinate system at every node in Phase 1 of this stage (like in the clusterless networks), only the master nodes compute their coordinate systems – the same as CRCE.

When Phase 1 is completed, all master nodes have established local coordinates for their slave nodes, relative to themselves. They then send this to the slaves to update their positions. Border nodes will accept the coordinates from their master node with the lowest ID.

Four message types are declared for use in [37]:

- M1: enables neighbours to establish distances from the sender,
- M2: sent by slave nodes to masters with distances to the slave's one-hop neighbours,
- M3: sent by master nodes to update the positions of its slaves, and
- M4: contains information about coordinate transformations to be made by the master nodes.

Algorithm 4.3 gives the outline of the process that is performed in this method of node positioning. Yet again the *while* loop is observed in the algorithm. This results in an expensive iterative coordinate convergence process, also present in RC-less.

```
timer->init()
decrement(timer)
if (timer = 0 AND no M1 received)
  type = master
  broadcast(M1)
  wait(M2s) and form coordinate system
  broadcast(M3)
  while (master_nodeID != minimum)
    wait(updates from border nodes)
    if nodeID < nodeID of update
      recalculate coordinates and broadcast(M3)
    end if
  end while
else
  type = slave
  broadcast(M1)
  wait(M1s are received)
  transmit(M2) to master
    while (master_nodeID != minimum)
      wait(coordinates from master)
      update coordinates
      if Number of coordinates > 1
        type = border
        transmit(M4) to masters with higher nodeid
      end if
    end while
  end if
```

Algorithm 4.3 The R-less method.

4.4 Chapter summary

Chapter 4 describes the three methods against which CRCE are compared to. Table 4.1 gives the classification of all four methods in terms of the type of convergence and network topology they exhibit.

Table 4.1 Classification of Positioning Methods

Method	Convergence	Topology
RC-less	Iterative	Clusterless
C-less	Radial	Clusterless
R-less	Iterative	Cluster-based
CRCE	Radial	Cluster-based

The *while loops* in the algorithms of the RC-less and R-less methods (iterative), suggest that these methods are more expensive, in terms of message cost, than the C-less and CRCE methods, in which no loops are present (radial).

It was also shown that one can expect cluster-based network topologies to yield much less transmitted messages than their clusterless counterparts.

CHAPTER 5 RESULTS AND DISCUSSION

In this chapter, the four methods of coordinate establishment are compared in terms of message cost by means of numerical (Section 5.1) and simulation (Section 5.2) procedures. The results are discussed in Section 5.3.

Numerical equations were formulated by counting the transmitted messages in smaller networks, recognising a pattern and developing a formula that fits. Simulations results were then obtained to confirm the validity of the numerical results.

The setup of the comparison is described: a flat square area with an edge length of L units, without any obstructions, is considered. It is then assumed that the area is covered with a uniform distribution of stationary nodes with density λ nodes/unit^2. Each node has a transmission range of r units and accurate distance measurements are provided.

Results were obtained for each of the two stages of every method and are presented next.

5.1 Numerical analysis

When one considers the number of messages transmitted by each node due to its location in the network, three different geographical types of nodes can be classified:

- Edge nodes – nodes located on the edge of the network and can be considered as the network boundary,
- Corner nodes – nodes located at the four corners of the rectangular network, and
- Inside nodes – those nodes contained within the network boundary.

Each of these nodes transmits a different number of messages throughout the coordinate establishment process. Large networks with many nodes will contain substantially more *inside nodes* than any other type of node. A square region (such as the one considered here) will, for example, have only four corner nodes. This is accounted for in the various formulas that follow.

5.1.1 Stage 1: Neighbour discovery

We consider the operation of the methods during Phase 1 and 2, with regards to the number of messages being transmitted, to derive formulas for message cost in Stage 1. The two clusterless methods operate in exactly the same way in this stage; as do cluster-based methods.

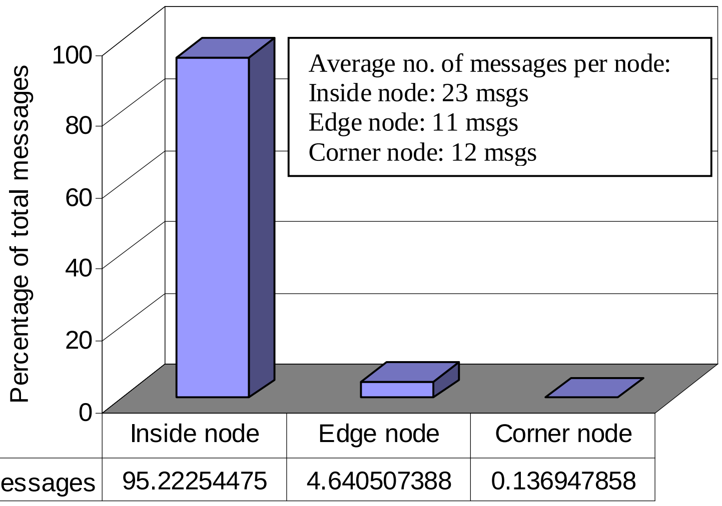
In Phase 1, the clusterless and cluster-based methods all transmit messages from each node to all of their neighbours. Therefore, the message cost for Phase 1 could simply be the product of the number of nodes in the network (N_T) and the number of neighbours for each node; but not all nodes in the network have the same number of neighbours – e.g. nodes located at the corners of the square network will have less neighbours than those in the middle, resulting in less transmissions. The same goes for border nodes. To accommodate this in the message cost formulas, it is necessary to differentiate between nodes based on their geographical locations in the network (edge, corner and inside nodes).

In Phase 2, the clusterless methods reply from each node to all its neighbours – the same amount of messages as in Phase 1. The message cost in stage 1 of the clusterless methods is computed by Equation 5.1.

$$M_{Clusterless} = 2\left[N_I.n_I + N_E.n_E + N_C.n_C\right], \tag{5.1}$$

where M is the total number of transmitted messages (message cost), N_I, N_E and N_C are the number of inside nodes, edge nodes and corner nodes, respectively. n_I, n_E and n_C are the number of neighbours for the respective node types, indicated by the subscripts.

Figure 5.1 shows the percentage of message contribution for the different geographical nodes at a density of 1 node/m^2.



	Inside node	Edge node	Corner node
% of messages	95.22254475	4.640507388	0.136947858

Area: 40m x 40m, r = 2.0m, λ = 1

Figure 5.1 Average number of message transmission for the different types of geographical nodes in a clusterless network setup

As expected, Figure 5.1 suggests that the majority of communication takes place on the inside of the network. These nodes also transmit twice as much messages as the edge and corner nodes.

Equation 5.2 calculates the message cost for the cluster-based methods. Unlike the clusterless methods, the cluster-based methods reply only from the slave/border nodes to the masters (which are established in Phase 1).

$$M_{Cluster-based} = \left[N_I.n_I + N_E.n_E + N_C.n_C\right] + N_K.n_I\,, \tag{5.2}$$

where N_K is the number of clusters in the network.

We now describe how to calculate the values for the different variables in (5.1) and (5.2).

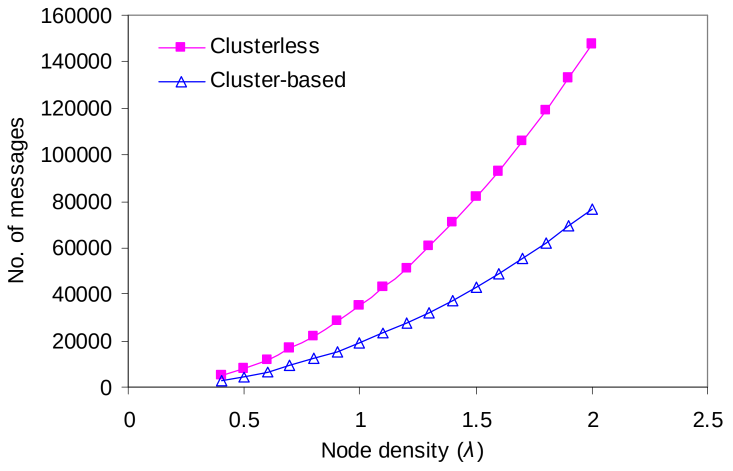
Given: transmission range (r), and number of neighbours for an inside node (n_I) – i.e. the number of neighbours for a node with no connection limitation due to its geographical location (e.g. a corner has connections only in one quadrant of its coverage area).

Calculate the node density (λ) according to (5.7), letting $N_I = n_I + 1$. Choose the number of nodes in the network. Keep in mind that the simulation setup states a square network; an intelligent number of nodes has to be chosen to keep the node count for N_E integer (in the next step), i.e. $N_T = i^2$, where $i = 1, 2, 3, \ldots$ The area of the network will then be $A = \lambda N_T$.

Now calculate the number of nodes for each of the three types of geographically classified nodes. According to the simulation setup, we have $N_C = 4$, $N_E = 4\left(\sqrt{N_T} - 2\right)$, $N_I = N_T - \left(N_E + N_C\right)$, and $N_K = A/\left(pr^2\right)$.

Then set the neighbour counts, n_I, n_E and n_C. n_I was given (or calculated in the first step if λ was given, by rearranging (5.7) to solve for n_I). n_E and n_C were obtained by setting up a number of networks with varying densities; λ = 0.4 to 1.8. By counting the number of neighbours for the corner and edge nodes in these networks and plotting them on a graph as a function of λ, it was possible to establish $n_E = 7.02(\lambda)$, and $n_C = 4.31(\lambda)$.

Figure 5.2 shows the number of transmissions in stage 1 for both the clusterless and cluster-based methods as a function of λ. The number of nodes in the network is kept constant (N = 1600 nodes).



Area: 40 m x 40 m, r = 2.0 m

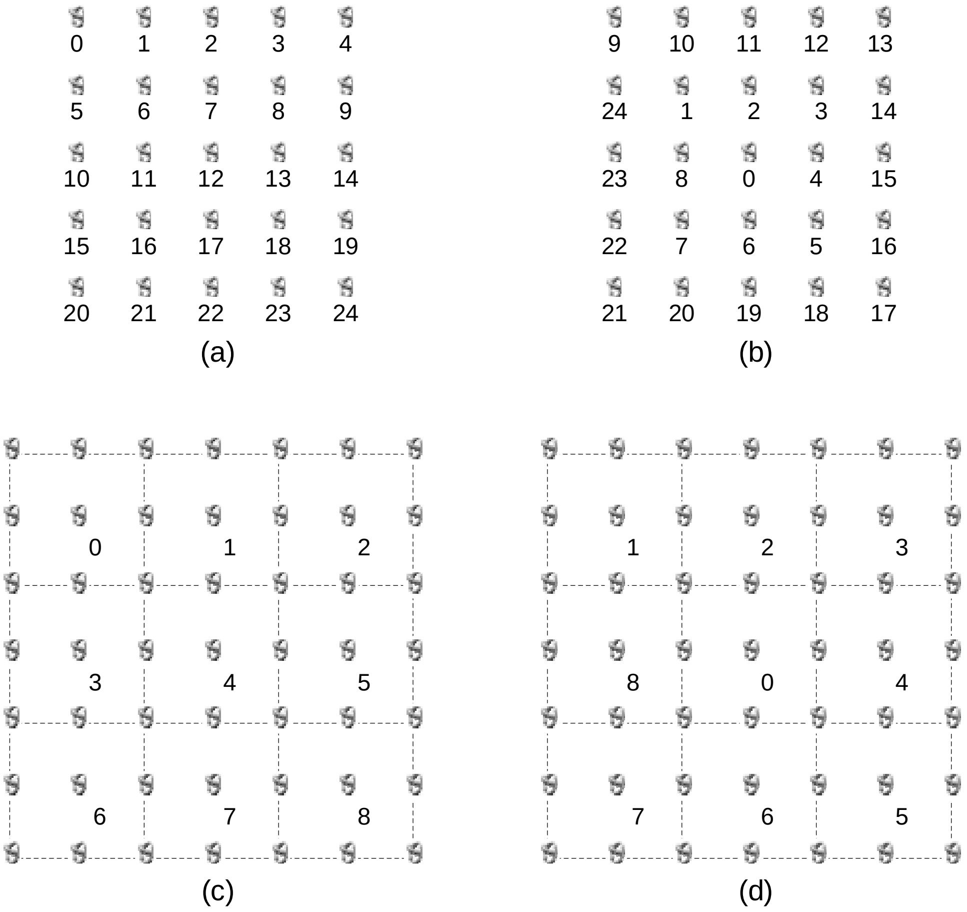
Figure 5.2 Number of messages transmitted during the neighbour discovery stage in a network of 1600 nodes

Note that the non-cluster based methods are at average 1.8 times as expensive as their cluster-based counterparts in Stage 1.

5.1.2 Stage 2: Coordinate establishment

The number of messages transmitted depends, in part, on the location of the sink node (radial methods) or the location of the node with the lowest ID (non-radial methods). This is because all position coordinates have to be calculated relative to the sink or the node with the lowest ID, leaving that node as the origin of the network-wide coordinate system.

Having the origin somewhere in the middle of the network shortens the convergence distance – distance between the origin and the furthest node in the network – as opposed to being located on the edge of the network. This affects only the coordinate establishment stage (Stage 2) and is considered in a *worst-case*, *best-case* fashion.



(a) (b) (c) (d)

Figure 5.3 (a) Clusterless worst-case node numbering, (b) clusterless best-case node numbering, (c) cluster-based worst-case numbering, and (d) cluster-based best-case numbering

Node numbering also influences message cost. It is particularly of interest at the numerical analysis because we need a consistent node numbering scheme in order to derive a formula for message cost calculation. The worst-case and best-case node numbering conventions for the clusterless methods are shown in Figure 5.3(a) and (b) respectively, and for the cluster-based methods in (c) and (d). Worst case scenarios are numbered from left to right, top to bottom and best case scenarios clockwise, concentrically from the middle outward.

5.1.2.1 Worst-case coordinate establishment

Geographically speaking, the sink node (radial methods) or the node with the lowest ID (non-radial methods), is now classified as a corner node.

Phase 1 of Stage 2 is concerned only with coordinate calculation and adds nothing to the message cost of the various localisation methods. All transmissions in Stage 2 take place during global coordinate convergence (Phase 2).

We describe the principle of deriving mathematical formulas for the message cost in Stage 2:

1. Set up relatively small networks, $5^2 \leq N_T \leq 15^2$, with a node density that will enable simple neighbour connections (λ = 0.716 results in $n_I = 8$; easy to draw network).
2. Identify the nodes with different message transmission numbers. This can be based on geographical location (as in Stage 1) and/or network role (slave, border and master). For clusterless networks we still consider only corner, edge and inside nodes, but for cluster-based methods we classify corner nodes, two types of edge nodes (slave nodes and border nodes with two masters), and three types of inside nodes (master nodes, border nodes with only two masters, and border nodes with four masters).
3. Physically count and record the number of message transmissions for every type of node (identified in step 2) by following the algorithm for each localisation method in Phase 2 of the coordinate establishment stage.
4. Identify a pattern from this. Simplify and formulate it into equations so that it can be applied to networks of any size.

Equations 5.3 and 5.4 calculate the number of message transmissions of the RC-less and C-less methods respectively.

$$M_{RC-less} = 0.952 n_I \sum_{i=0}^{\sqrt{N_T}-1} \left(N_T - i^2 \right), \tag{5.3}$$

$$M_{C-less} = N_I . n_I + N_E . n_E + N_C . n_C . \tag{5.4}$$

Figure 5.4 illustrates the worst-case coordinate establishment for the clusterless methods, in terms of message cost. It gives evidence to prove the enormous impact that radial convergence has on coordinate establishment even in non-cluster based methods. The poor performance of the RC-less method is attributed to its inherent iterative property (5.3).

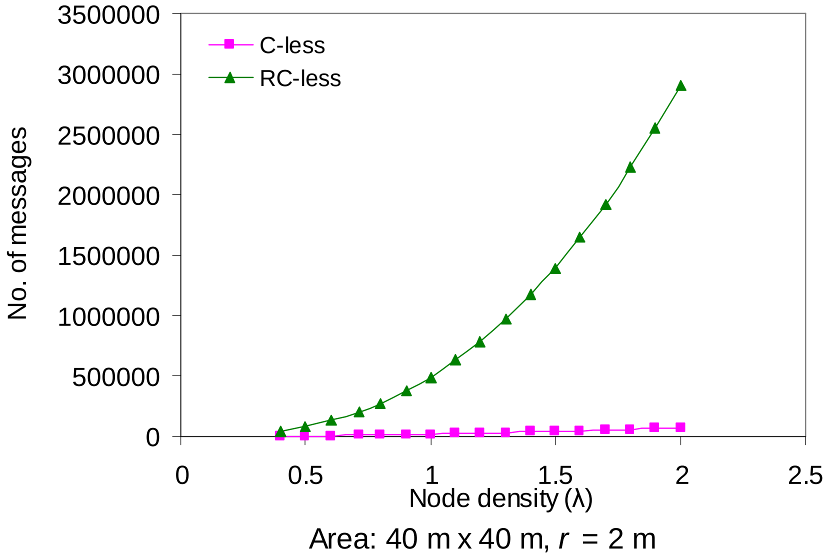


Figure 5.4 Worst-case, second stage, clusterless message transmission

The two cluster-based methods are compared next. This time the density is kept constant and the number of message transmissions is calculated as a function of the total number of clusters (N_K) in the network (5.5), (5.6). λ is chosen such that the number of slave nodes in each cluster equals an arbitrarily chosen number of neighbours, 8.

$$M_{R-less} = \sum_{i=0}^{\sqrt{N_K}-2} \left(N_I - 2i\sqrt{N_C}\right) + \sum_{i=0}^{\sqrt{N_K}-2} \left(N_E - 4i\right) + 3\sum_{i=1}^{\sqrt{N_K}-1} i^2 + 4 \sum_{i=0}^{\substack{\sqrt{N_K}\, if\, (\sqrt{N_K}-1) \bmod 2 = 0 \\ (\sqrt{N_K}-1)\, if\, (\sqrt{N_K}-1) \bmod 2 \neq 0}} i\left[\sqrt{N_K} - (i+1)\right], \quad (5.5)$$

$$M_{CRCE} = 8N_K + 3\left(\sqrt{N_K} - 1\right)^2 + 2\left(\sqrt{N_K} - 1\right) \quad (5.6)$$

The iterative property of RC-less is again evident in (5.5). Figure 5.5 shows the effect this has on the number of transmissions in the cluster based coordinate establishment. It is observed that that CRCE scales linearly as the number of clusters in the network increase.

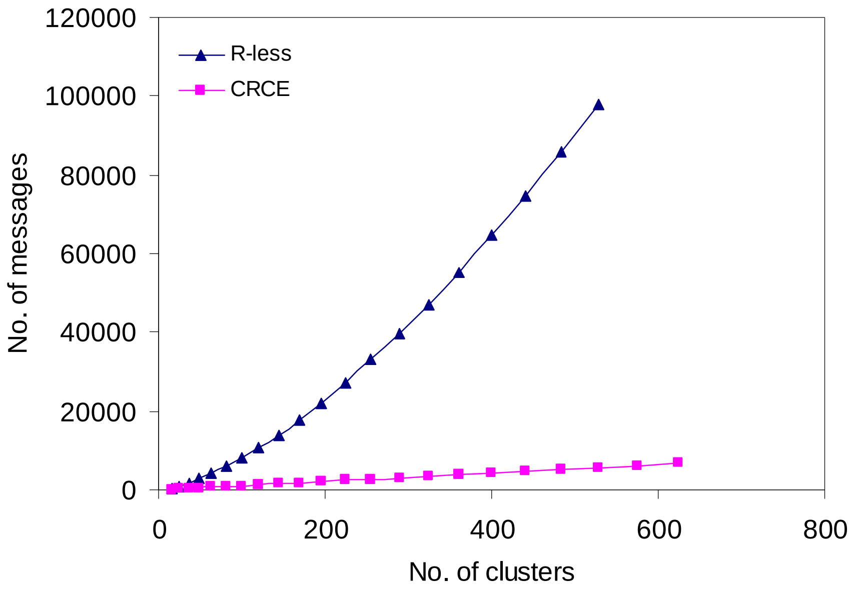
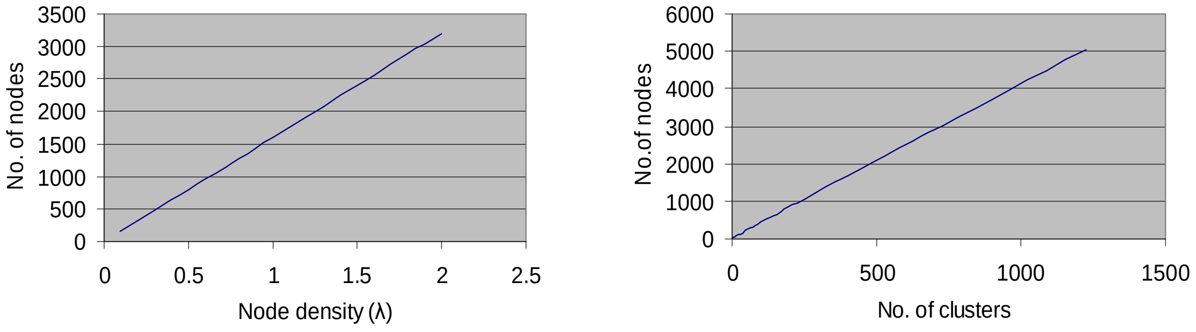


Figure 5.5 Worst-case, second stage, cluster-based message transmission

For the clusterless methods, the density of the nodes is varied in a set area of 1600m^2, and for the cluster-based methods the number of clusters is varied, limited in size by the number of nodes these clusters will allow (i.e. the size of the cluster-based network is directly proportional to the number of clusters in the network).

To compare all four methods of coordinate propagation on one bar graph (to level the plane), λ of the clusterless methods has to be set to a value that corresponds to the density of the cluster-based methods. The number of clusters is then set to a value that will result in a match between the numbers of nodes in each network at the chosen λ. Figure 5.6 shows how the number of nodes increases as λ or the number of clusters increase.



(a) (b)

Figure 5.6 Increase in the number of nodes in the network as (a) the node density and (b) the number of clusters increase

The degree of connectivity in a WSN refers to the average number of neighbours each node should have. It has been proven in [49] that a network needs a degree of about 6 in order to have complete connectivity with high probability.

Because the connectivity constraint state that the number of neighbouring nodes in the domain of node *i* should preferably be greater than six, we choose a safe degree of connectivity of eight ($D_i = 8$). Therefore the total number of nodes in the domain of *i* is $N_i = 9$ (including node *i*).

The density of a uniformly distributed network can be calculated as a function of N_i and a given value for the transmission range *r*, as follows:

$$l = \frac{N_i}{pr^2}. \tag{5.7}$$

According to (5.7), if the transmission range is chosen such that $r = 2$m, and $N_i = 9$ nodes, then $\lambda = 0{,}716197244$ nodes/m^2. From Figure 5.6a the number of nodes in the network is indexed to be $N_T = 1146$ nodes. This is indexed again on Figure 5.6b to get 260 clusters.

Now it is possible to compare the number of messages of the clusterless methods at λ = 0,716197244 nodes/m^2 to the cluster-based methods with the number of clusters set to 260. Figure 5.7 gives the comparison.

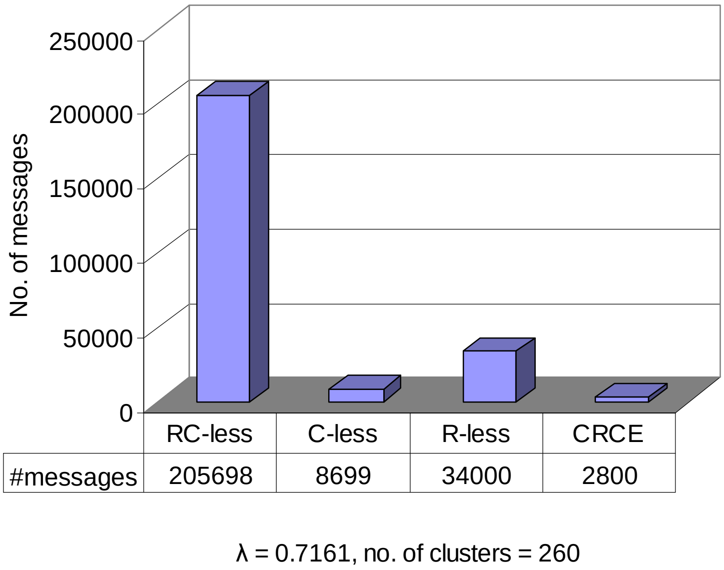


Figure 5.7 Worst-case, second stage, message transmissions

The superior performance of the CRCE method is observed. The performance of the C-less method over that of the cluster-based R-less method is also noted. This is due to the inefficient iterative coordinate propagation of the R-less method.

5.1.2.2 *Best-case coordinate establishment*

The sink node (radial methods) or the node with the lowest ID (iterative methods) is assumed to be located in the middle of the network. The node numbering is clockwise, from the centre, concentrically outward as shown in Figure 5.3b and 5.3d for clusterless and cluster-based topologies, respectively. This numbering scheme does not make it possible to deduct a formula for message cost for all the methods – especially the clusterless methods.

For the R-less method an educated guess of the message cost can be made. It is explained. Figure 5.8 presents an example network of 9 clusters with D = 8.

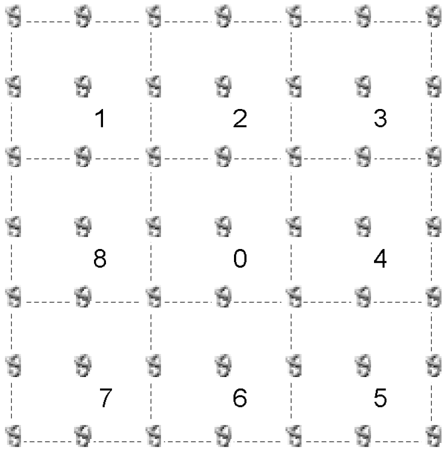


Figure 5.8 9 clusters in the best-case numbering convention at *D* = 8

Every node in this network contributes to the total number of transmitted messages. A message cost equation can be formulated for the master nodes (cluster heads) in the network, by counting the number of transmitted messages for small networks and then recognising a pattern. This is given in the form of (5.8). Again the iterative property of R-less is evident from the sigma in the equation:

$$M_{CH,R-less} = N_K + \sum_{i=0}^{(\sqrt{N_K}-1)/2} \left[N_K - (2i-1)^2 \right]. \tag{5.8}$$

The number of messages transmitted by the other nodes (non-master nodes) in the network is counted for four small networks of 9, 25, 49 and 81 clusters. The results are given in Table 5.1.

Table 5.1 Number of messages from non-master and master nodes

No. of clusters	**Number of messages from non-master nodes**		**Percentage of Total messages**	
	Non-masters (Count)	Masters (Eq. 5.8)	Non-masters	Masters
9	32	136	19.05%	80.95%
25	158	520	23.30%	76.70%
49	433	1288	25.16%	74.84%
81	931	2568	26.61%	73.39%

It is assumed that the trend in the number of messages transmitted for the non-master nodes will continue in much the same way if the number of clusters increases. Therefore, the next four points (number of clusters = 121, 169, 225 and 289) in Figure 5.9 can be fitted/fixed to resemble a continuous graph with no sharp changes.

This gives an estimate of message cost for the non-master nodes. Because a formula for message cost of the master nodes is available (5.8) – i.e. accurate message cost – and because these nodes account for roughly three quarters of the total number of messages in the network (Table 5.1), a small error in the message cost estimation of the non-master nodes will have no significant impact on the total number of messages transmitted.

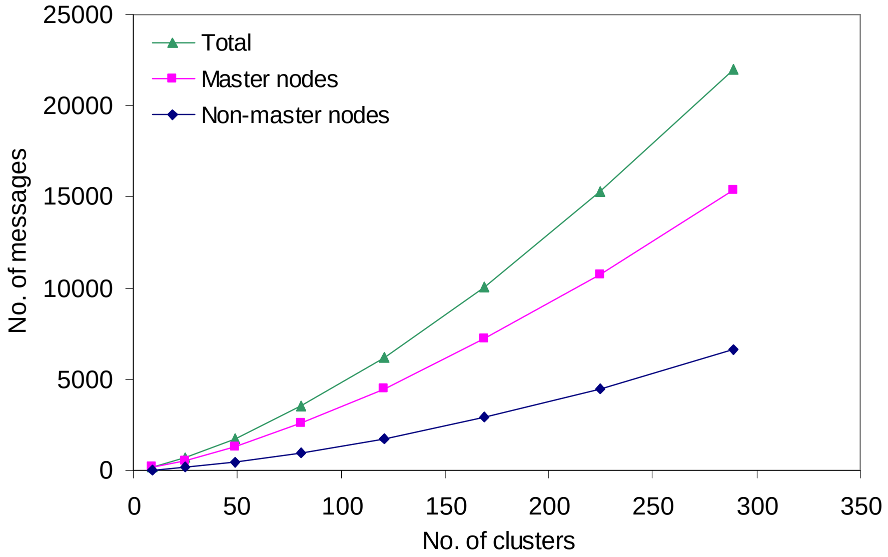


Figure 5.9 Estimated best case message cost for the R-less method

The formula for message cost for CRCE is given in Equation 5.9 and is compared to the R-less method in Figure 5.10. When it is kept in mind that $n_K = 8$, then

$$M_{CRCE} = 3(\sqrt{N_K} - 1)^2 + 8N_K\,. \tag{5.9}$$

N_K is once again the number of clusters in the network.

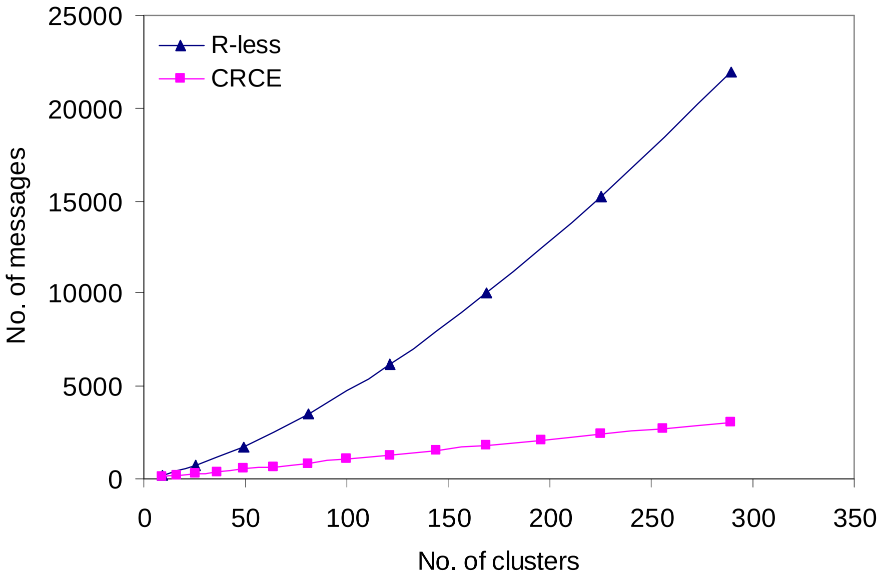


Figure 5.10 Best-case, second stage, cluster-based message transmission

Again it is observed that CRCE scales linearly as the number of clusters increase. The worst and best-case performances of the cluster-based methods are compared in Figure 5.11 and in Figure 5.12 at 256 clusters.

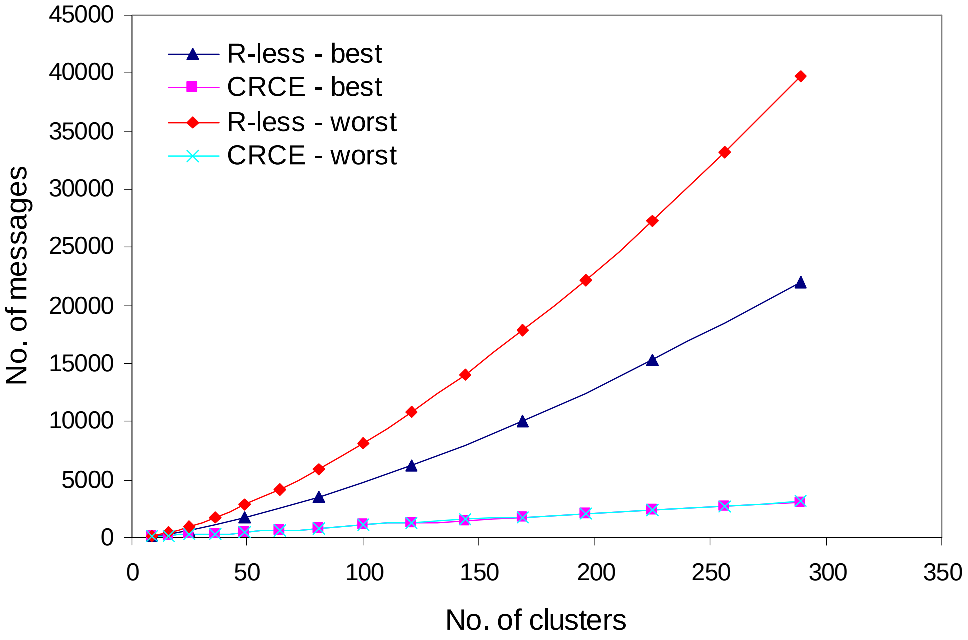
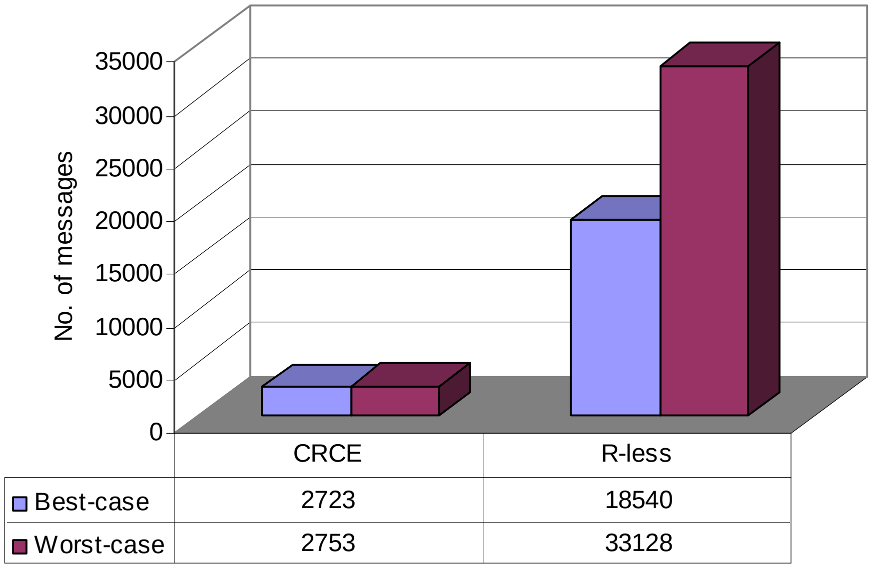


Figure 5.11 Best and worst-case performances of the cluster-based methods, $D = 8$



	CRCE	R-less
■ Best-case	2723	18540
■ Worst-case	2753	33128

Figure 5.12 Best and worst-case performances of the cluster-based methods at 260 clusters, *D* = 8

The placement of the sink node in CRCE does have almost no affect on the message cost. In contrast to this, sink placement in the R-less method can influence the message cost by up to 45%.

5.2 Simulation

Simulation results were obtained by implementing all four algorithms in OMNeT++ [50] with the Mobility Framework (MF) [51]. OMNeT++ is an open source object-oriented modular discrete event simulator and stands for Objective Modular Network Testbed in C++. MF t is a framework to support simulations of wireless and mobile networks within OMNeT++. The simulations were consequently programmed in C++.

5.2.1 Stage 1: Neighbour discovery

Figure 5.13 compares the message cost of the neighbour discovery stage. Again the cluster-based methods transmit the same number of messages, and so do the clusterless methods. The numerical results are also presented on the graph.

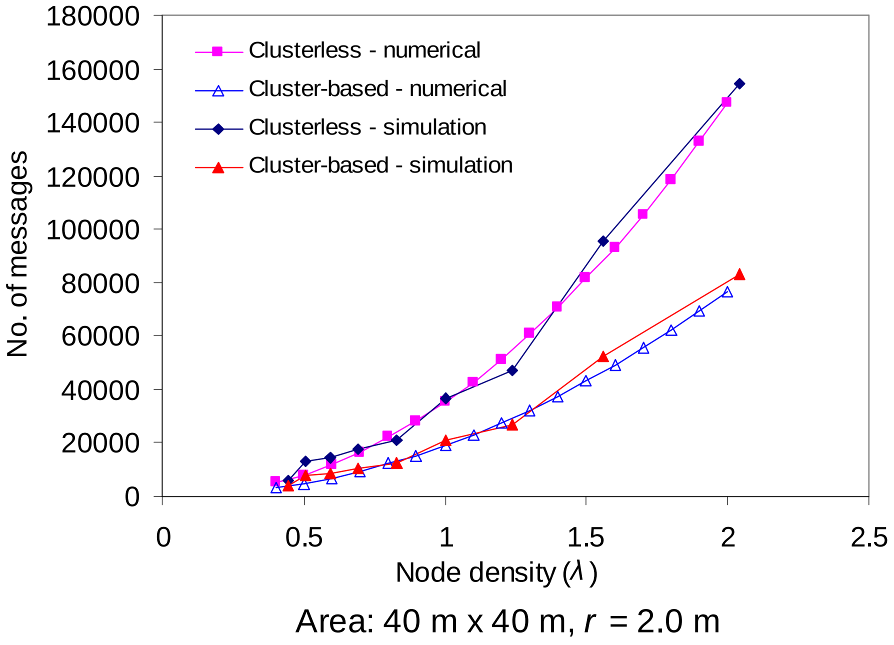


Figure 5.13 Number of messages transmitted during the neighbour discovery stage – simulation and numerical

It is observed that the simulation results are much the same as the numerical ones, and therefore validate the numerical formulas (5.1) and (5.2) as approximations to message cost in the neighbour discovery stage.

5.2.2 Stage 2: Coordinate convergence

The worst-case placement of the sink node or the node with the lowest ID in the network is considered, followed random node numbering to see the effect it has on the message cost of the four different coordinate establishment methods in Stage 2.

5.2.2.1 Worst-case sink/lowest ID node placement

Figure 5.14 compares the worst-case simulation results with the numerical results for the clusterless methods.

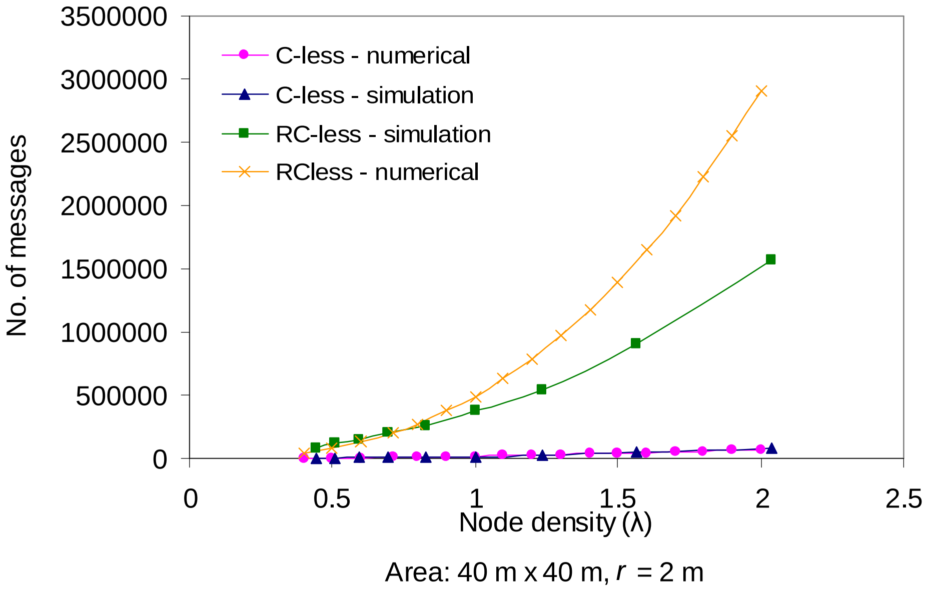


Figure 5.14 Numerical and simulation worst-case message cost for the clusterless methods

The RC-less method in Figure 5.14 scales the C-less method too small to make any worthwhile observations. C-less is presented on its own in Figure 5.15.

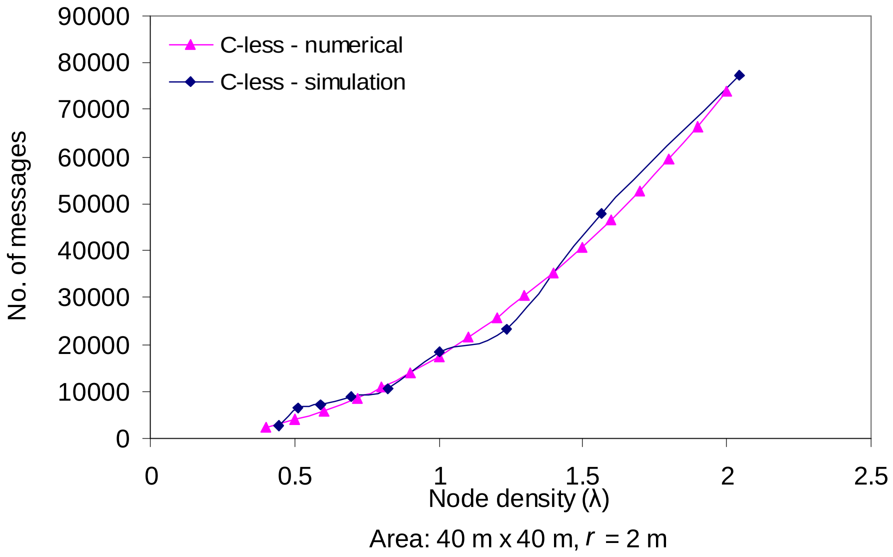


Figure 5.15 Numerical and simulation results for worst-case message cost of the C-less method

From Figure 5.15 it is evident that (5.4) is a valid formula for computing message cost of the C-less method in Stage 2. Figure 5.16, however, suggests that the difference between the numerical and simulation results increases as the density of the network increases. It is explained next.

As the density of the network increases, more and more nodes close to the edge of the network do not obtain the maximum number of neighbours that are possible for a given transmission range. The ***maximum number of neighbours*** can be defined as the number of nodes in the domain of node *i*, where node *i* is located such that the number of its neighbours are only limited by the transmission range of node *i*, and not by a lack of nodes outside of the deployment area. Figure 5.16 depicts this.

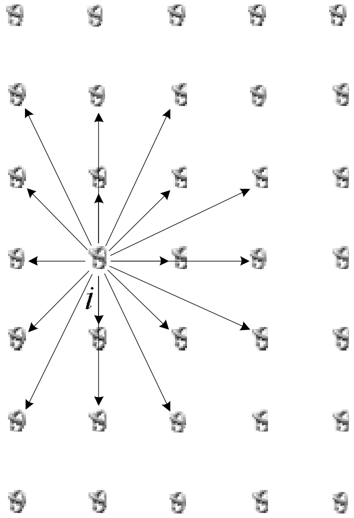


Figure 5.16 Limited number of neighbours on the left side of node *i*

The formulas for message cost do not take this into account. It assumes that all nodes within the boundary of the network have maximum neighbour connectivity.

The reason why it is only visible in the RC-less method is that the error compounds with every iteration of the convergence process. In a square network of 40 x 40 nodes, density of 1 node/m^2 and transmission range of 2m, RC-less iterates 40 times and C-less only once. This results in a higher numerical message cost than the actual simulation cost.

Figure 5.17 presents the simulation and numerical message costs for the cluster-based methods.

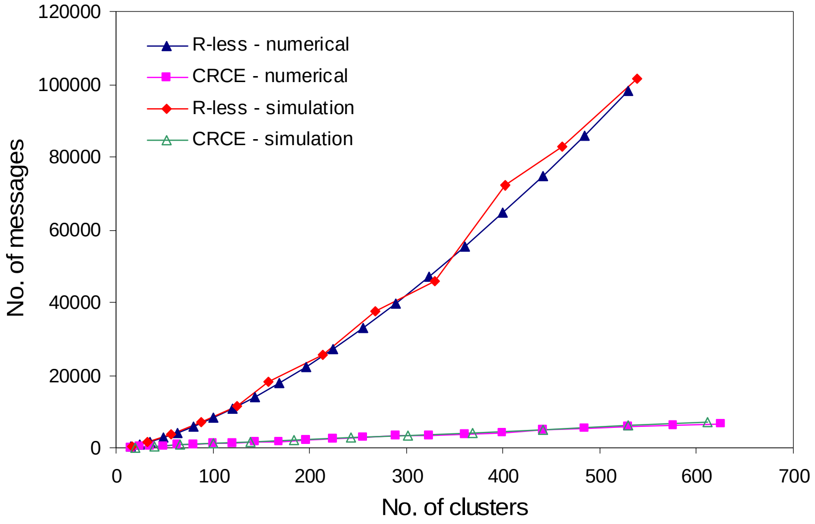


Figure 5.17 Numerical and simulation worst-case message cost for the cluster-based methods

CRCE is scaled too much to compare the numerical and simulation results effectively. Once again, it is displayed separately in Figure 5.18.

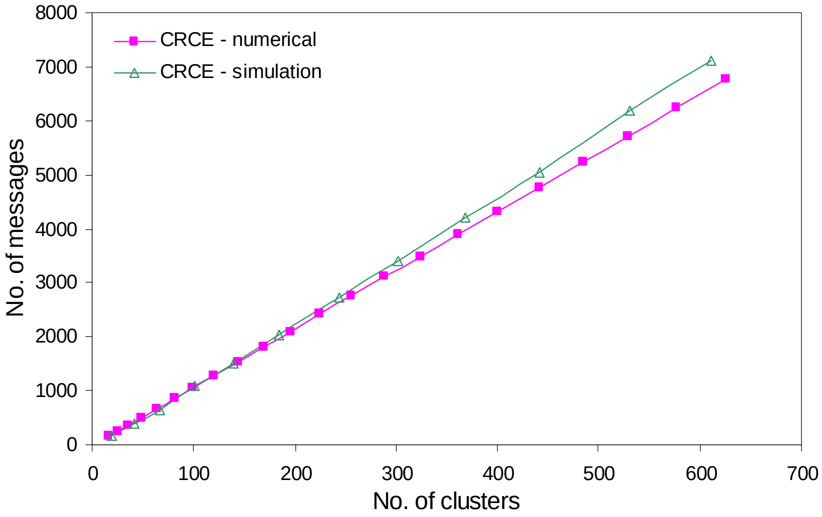


Figure 5.18 Numerical and simulation results for the message cost of CRCE

The correlation between the numerical and simulation results is clearly visible. Furthermore, CRCE is shown to scale linearly as the clusters in the network increase. (5.5) and (5.6) are now validated.

5.2.2.2 Random sink/lowest ID node placement

In practice, there will often be no control over the location of the sink node (radial methods), or the node with the lowest ID (iterative methods), within the network – sensor nodes may be dropped from a plane or a handful may be tossed over the area of interest.

This is investigated to find the effect it has on the message cost of the different methods. It has to be kept in mind that this applies only to Stage 2, the coordinate establishment stage. The number of messages in Stage 1 is not influenced by the position of the sink node or the node with the lowest ID.

The simulation was performed 20 times for each method with a different uniformly distributed random placement of nodes for every run. We want to compare the cluster-based and clusterless methods against each other, thus, the density and the number of nodes for these networks have to be the same. The following values were chosen in the same manner as in Section 5.1.2.1; λ = 0,716197244 nodes/m^2, r = 2m and N_T = 1146 nodes. Mean values for message cost were obtained from the simulations and are displayed in Figure 5.19.

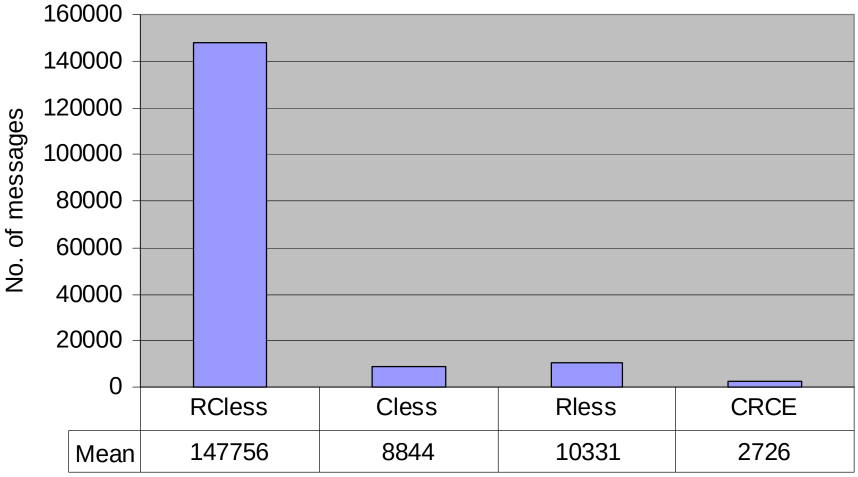


Figure 5.19 Mean values of message cost for random node placement

The variation in message transmissions for each method is normalised relative to the mean of that method in Figure 5.20.

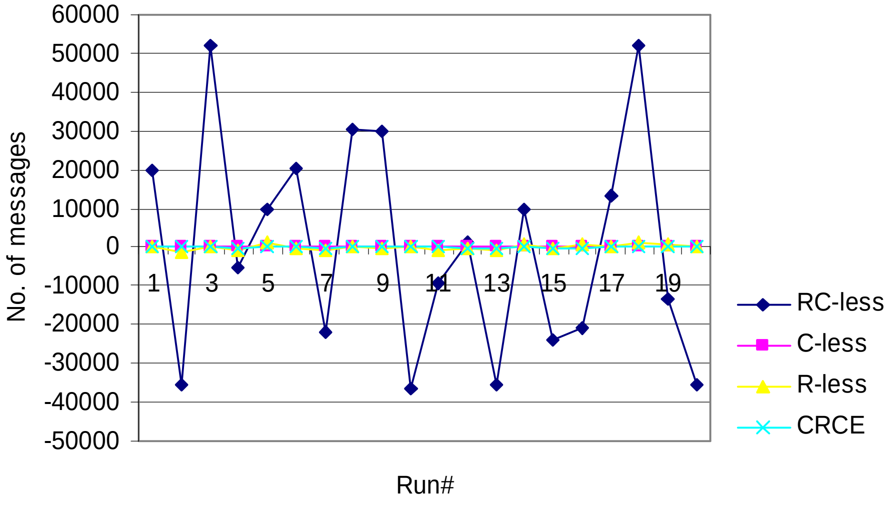


Figure 5.20 Variation of message transmission with random placement of nodes in the same network

The huge variation in message count for the RC-less method scales the rest of the methods too small to make any sense from it. The graph is redisplayed in Figure 5.21, but without RC-less.

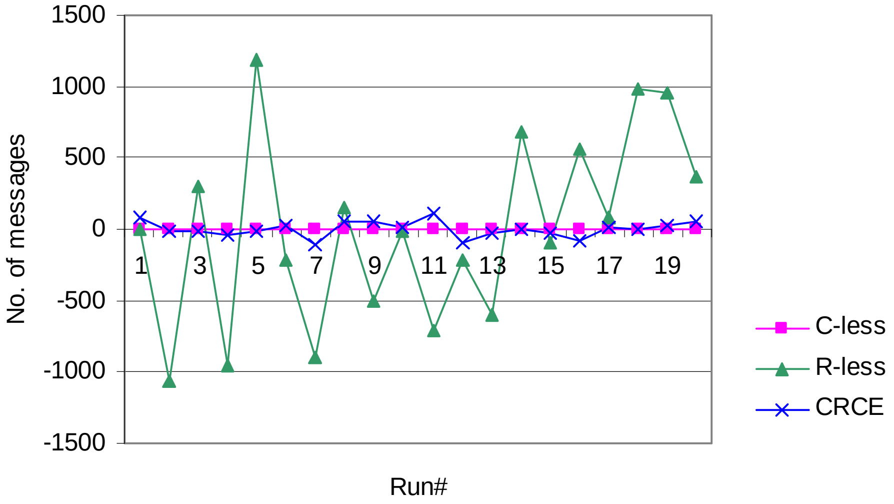


Figure 5.21 Variation of message transmission with random placement of nodes in the same network – without RC-less

It is clear that node placement in the network influences the message cost of the iterative methods (i.e. the non-radial methods) much more than the radial methods – especially for RC-less. C-less is not influenced at all by random node placement and the little variation in CRCE is due to different cluster formations in each run.

5.3 Stages combined

Finally, an overall comparison (Stage 1 and 2) is made for all four coordinate establishment methods.

Again we set λ = 0,716197244 nodes/m^2, r = 2m and N_T = 1146 to obtain a common ground for the clusterless and cluster-based methods regarding node density and number of clusters respectively.

The result is shown in Figure 5.22 and again in Figure 5.23 without RC-less.

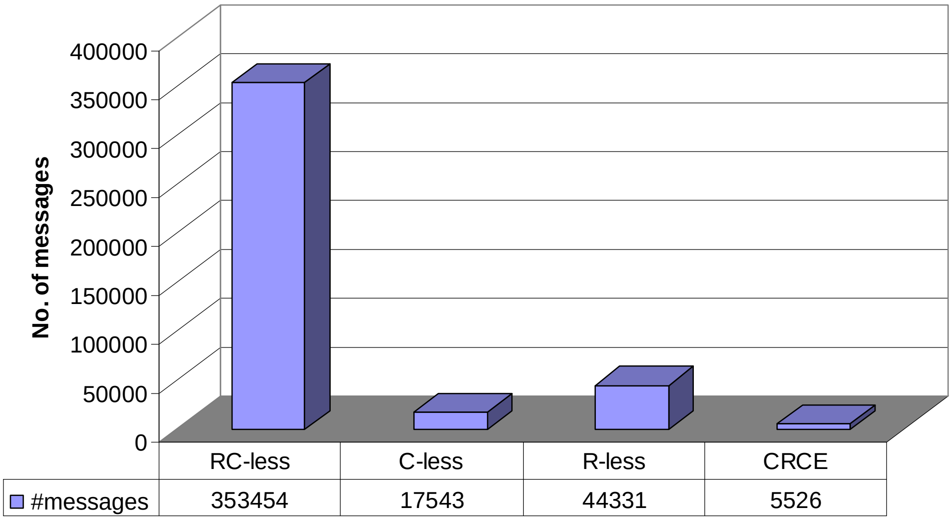


Figure 5.22 Total message cost for the four coordinate establishment methods

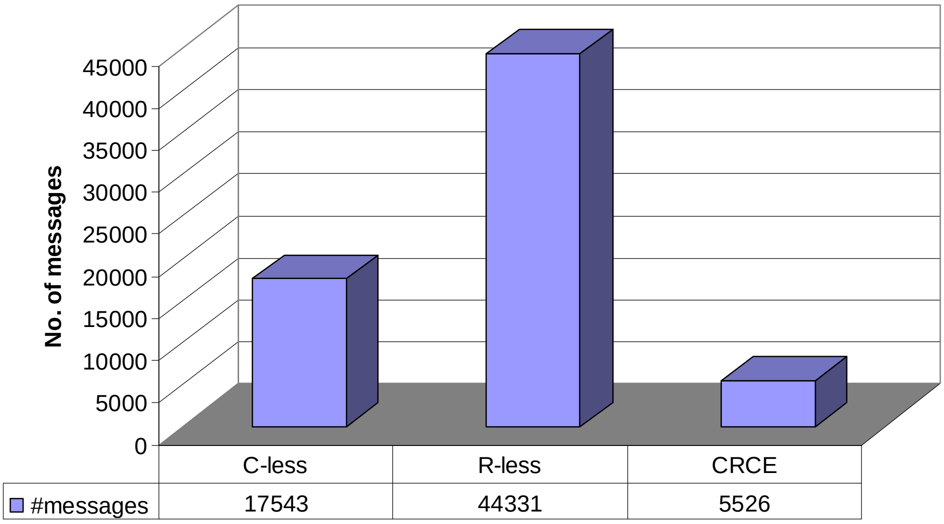


Figure 5.23 Total message cost for the coordinate establishment methods – omitting RC-less

CRCE outperforms its closest competitor, C-less, by 68.5%, and R-less by 87.5%.

5.4 Discussion

The interpretation of the figures/graphs is given throughout the previous two sections. The key observations are discussed here.

In Stage 1 of the coordinate establishment process, the clusterless methods transmit an equal number of messages. This is due to the fact that the same neighbour discovery process is used in both: all nodes record one-hop information, then all nodes transmit one-hop information. The same is true for the cluster-based methods: all nodes record one-hop information, all slave/border nodes transmit one-hop information to their respective master(s) only.

Because cluster-based methods do not broadcast messages to all nodes in Stage 1 (they only send to master nodes), they cost an average of 45% less than their more expensive clusterless counterparts.

Random node numbering and sink/lowest ID node location in the network affects only the coordinate establishment stage (Stage 2) of the localisation process. This is because Stage 1 is not concerned with the distance to the node with the lowest ID. The convergence distance determines the number of iterations it will take to get a network-wide coordinate system.

Note also that it doesn't influence the radial methods either, as they perform only one iteration no matter where the sink node is located. In contrast, the difference in message cost can be as much as 45% (difference between best and worst-case).

The small variations in CRCE are attributed to the fact that the algorithm forms new clusters in each run, which is dependant on random timers. The number of clusters in each network then differs slightly and so does the message cost.

Simulation results are shown to validate the numerical evaluation formulas with relatively high accuracy, except for the case of RC-less in Stage 2. Here, the message cost using numerical analysis increase too fast with respect to an increase in density, due to more nodes that are closer to the edge of the network as the number of nodes in the network increases. The formulas for message cost do not take this into account.

The reason why it is only visible in the RC-less method is that the error compounds with every iteration of the convergence process, and RC-less iterates by far the most of all methods (40 times in a 40 x 40 nodes network compared to 1 iteration for CRCE and C-less in any size network – radial property).

In Stage 2, by setting up a network with the following characteristics, the average message cost per node can be obtained for all the methods: λ = 0,716197244 nodes/m^2, r = 2m and 1146 nodes in the network.

Table 5.2 Average message cost per node for Stage 2 at λ = 0,716197244 and N_T = 1146

Method	No. of messages per node
RC-less	128.9319372
C-less	7.717277487
R-less	9.014834206
CRCE	2.378708551

CRCE is by far the cheapest relative coordinate establishment method of them all and is shown to exhibit a linear scaling ability in Stage 2. What is interesting to note in Table 5.2 is that the radial C-less method outperforms the cluster-based R-less method in Stage 2.

This trend is also present in the overall comparison of the localisation methods and suggests that radial convergence alone is more effective than implementing only clustering in a network. Not only is it cheaper on messages but also simpler in implementation and less resource intensive.

By implementing radial convergence (proposed in this research) in a cluster-based topology, CRCE is shown to be almost 70% more effective in message cost reduction than the modified RC-less method, C-less, and close to 90% more effective than the published R-less method.

CHAPTER 6 CONCLUSION

6.1 Research summary

The problem of scalable, energy efficient, localisation in Wireless Sensor Networks was addressed by developing a node positioning method called Cluster-based Radial Coordinate Establishment (CRCE) with the following two characteristics:

- *Localised distributed processing*

  Instead of having one powerful node performing complicated, processor intensive position computation for the entire network, the potential of geographically distributed processors was utilised to reduce the message overhead that these types of methods incur.

  A cluster-based network topology was implemented that made it possible to reduce the number of message transmissions during the coordinate establishment process significantly.

  Scalability in the sensor network was achieved by forming local coordinates at only a small subset of the nodes – cluster heads (or master nodes).

- *Relative coordinate system*

  It is unrealistic to assume the presence of nodes with a-priori location knowledge (beacon or landmark nodes) in rapidly deployed sensor networks, therefore the network coordinate system resulting from CRCE is converged to a local reference system (sink cluster) that is arbitrary in direction, but still results in network-wide coherence.

  Coordinate updates are used to converge all local systems to this network-wide relative coordinate system and traverse the network radially, rather than iteratively. This reduced the message cost even more.

6.2 Results summary

Numerical simulations were used to obtain formulas for message cost calculation in coordinate establishment for WSNs. These formulas depend heavily on node numbering conventions and are restricted to clusterless/cluster-based and iterative/radial localisation methods.

Simulations were performed to validate these formulas and proved that both radial convergence and clustering topologies cause a reduction in message cost when implemented in separate networks.

CRCE combines both of these to form one localisation method which is shown to perform almost 90% better with regards to message cost, than its closest published competitor.

The combination of the two results in a more complicated algorithm though, and therefore it would need a bit more resources, in terms of memory, compared to the other methods.

6.3 Final conclusion

Finally, the hypothesis posed in Section 1.5.1 can now be evaluated to conclude this research document. It is restated below:

> “Coordinate establishment in WSNs is more expensive, in terms of message cost, when the network topology is clusterless and coordinate propagation is done in an iterative manner, as opposed to a cluster-based, radial coordinate convergence method”.

CRCE is a cluster-based, radial coordinate establishment method that is shown to outperform the iterative and/or clusterless methods by 68.5% to 87.5%. Thus, the hypothesis is *accepted*.

6.4 Research contribution

This research demonstrates the benefits of radial convergence in both:

- Clusterless, and

- Cluster-based WSNs.

The graphs and figures can be used for message cost estimation and the mathematical models that were developed can be used for message cost calculation in:

- Clusterless and iterative methods,
- Clusterless and radial methods,
- Cluster-based and iterative methods, and
- Cluster-based and radial methods.

CRCE was developed, and is shown to be:

- scalable, and
- energy efficient.

It can serve as benchmark for other relative and distributive localisation methods whilst position centric routing algorithms can be employed on top of CRCE. Rapid deployment of sensor networks is also made possible because it does not make use of nodes with a-priori location information.

6.5 Future work

The convergence time of the coordinate establishment methods studied in this research could also be of interest to some applications. For most it would not be that important because the lifetime of the network will be much longer than the setup time and the phenomena would usually be measured over a long period of time rather than micro seconds after the deployment of sensors.

Optimal position refinement for the CRCE method is another area of interest. Erroneous distance measurements require subsequent algorithms to refine the position coordinates for wireless sensors. The best type of algorithms would also exhibit the distributive processing property.

An insight into the memory and processing power requirements would also be of interest to be able to identify sensors with adequate resources for the CRCE method.

Mobile nodes in the network will affect the message cost of CRCE negatively. More research is necessary to understand exactly how much it will be affected and under what conditions (percentage of mobile nodes, speed of movement, etc.). It is known, however, is that the message cost of the other methods will be affect proportionately in the presence of mobile nodes.

Finally, the implementation of CRCE in real sensors and test networks would yield the ultimate test of the newly proposed coordinate establishment method.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, Utah, May 2001*.

[2] J. Cox, "New frontier for wireless: Sensor networks," Network World, http://www.nwfusion.com/news/2004/0607sensors.html. Last accessed 15 January 2005.

[3] D. Niculescu, "Positioning in ad hoc sensor networks," *IEEE Network*, vol. 18, no. 4, pp. 24-29, July/Aug. 2004.

[4] "IEEE Std 802.15.4-2003, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Computer Society, http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf. Last accessed October 2004.

[5] IEEE 802.15 WPAN™ Task Group 2 (TG2), http://www.ieee802.org/15/pub/TG2.html. Last accessed October 2004.

[6] IEEE 802 LAN/MAN Standards Committee, http://www.ieee802.org.

[7] Broadband Radio Access Networks, http://portal.etsi.org/portal_common/home.asp?tbkey1=BRAN. Last accessed September 2004.

[8] 802.11 5GSG PAR and 5 Criteria, http://www.ieee802.org/17/email/msg00374.html. Last accsessed October 2004.

[9] ZigBee Alliance, http://www.zigbee.org.

[10] Wi-Fi Alliance, http://www.wi-fi.org.

[11] "Sensor network," http://encyclopedia.lockergnome.com/s/b/Sensor_network. Last accsessed March 2005.

[12] A. Savvides, C.-C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of Mobile Computing and Networking (MobiCom), July 2001*.

[13] M. Vossiek, L. Wiebking, P. Gulden, J. Wieghardt, C. Hoffmann, and P. Heide, "Wireless local positioning", *IEEE Microwave Magazine*, vol. 4, issue 4 , pp. 77 – 86, December 2003.

[14] Goniomtry web site, "Goniometry," http://academic.uofs.edu/faculty/kosmahle1/courses/pt350/goniomet/gonpage1.htm. Last accessed 12 January 2005.

[15] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," *WSNA'02,Atlanta, Georgia, USA, September 28, 2002*.

[16] J. Riba and A. Urruela, "A non-line-of-sight mitigation technique based on ML-detection," *ICASSP Montreal, Quebec Canada, May 2004*.

[17] D. Niculescu and B. Nath, "Ad-hoc positioning system (APS) using AOA," in *Proceedings of IEEE Infocom, San Francisco, CA, 2003*.

[18] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of IEEE INFOCOM '01, Anchorage, AK, Apr. 2001*.

[19] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28-34, October 2000.

[20] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," *MobiCom '03, San Diego, California, USA, September 14-19, 2003*.

[21] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," *In Mobile Computing and Networking*, pp. 32–43, 2000.

[22] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," *GLOBECOM, San Antonio, November 2001*.

[23] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Processing*, pp. 2137-2148, August 2003.

[24] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. S. J. Pister, "System architecture directions for networked sensors," *Operating Systems Review*, vol.34, pp. 93-104, Dec. 2000.

[25] R. L. Moses, D. Krishnamurthy, and R. Patterson, "An auto-calibration method for unattended ground sensors," in *Proceedings of ICASSP, vol. 3, pp. 2941–2944, May 2002*.

[26] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks, Berkeley, California, USA, April 2004*.

[27] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," *MobiHoc'03, Annapolis, Maryland, USA, June 2003*.

[28] C. Savarese, J. Beutel, and J. M. Rabaey, "Locationing in distributed ad-hoc wireless sensor networks," in *Proceedings of IEEE ICASSP, pp. 2037-2040, 2001*.

[29] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of the First ACM International Workshop on Sensor Networks and Applications (WSNA 2002), Atlanta, Georgia, USA , pp:112-121, September 2002*.

[30] D. Niculescu and B. Nath, "Position and orientation in ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 2, no. 2, pp. 133–51, April 2004.

[31] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," *GLOBECOM, San Antonio, 2001*.

[32] D. J. Dayley and B. M. Bell, "A method for GPS positioning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 1148-54, 1996.

[33] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C: the art of scientific computing*, 2nd ed., New York: Cambridge University Press, 1992.

[34] J. Zhang, K. Premaratne, and P. H. Bauer, "A distributed self-organization algorithm for ad-hoc sensor networks," *IEEE WCNC 2003, vol. 3, pp:1591 - 1596, March 2003*.

[35] J. Zhang, K. Premaratne, M. Dogruel, and P. H. Bauer, "Task-oriented self-organization of ad hoc sensor systems," in *Proceedings of IEEE SENSORS 2002, Orlando, Florida, USA, June 2002*.

[36] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *34th Hawaii International Conference on System Sciences (HICSS-34), Maui, Hawaii, pp. 3481-3490, Jan. 2001*.

[37] R. Iyengar and B. Sikdar, “Scalable and Distributed GPS free Positioning for Sensor Networks,” *IEEE ICC 2003,* May 2003.

[38] D. Niculescu and B. Nath, “Localized positioning in ad hoc networks,” *Elsevier's Journal of Ad Hoc Networks*, Special Issue on Sensor Network Protocols and Applications, vol. 1, issues 2-3, pp. 211-349, September 2003.

[39] K. Römer, “The lighthouse location system for smart dust,” *ACM/USENIX Conf. Mobile Sys., Apps., and Svcs., San Francisco, CA, pp. 15–30, May 2003*.

[40] G.G. Finn, *Routing and addressing problems in large metropolitan-scale internetworks*, ISI/RR-87-180, ISI, Mar 1987.

[41] L. Hughes, O. Banyasad, and E. Hughes, “Cartesian routing,” *In Computer Networks: The International Journal of Computer and Telecommunications Networking*, 34(3), pp. 455-466, September 2000.

[42] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, February 2003.

[43] I. F. Akyildiz and I. H. Kasimoglu, “Wireless Sensor and Actor Networks: Research Challenges,” *Ad Hoc Networks Journal (Elsevier)*, vol. 2, no. 4, pp. 351-367, October 2004.

[44] I. Stojmenović, “Position-based Routing in Ad Hoc Networks,” *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–34, July 2002.

[45] M.N. Halgamuge, S.M. Guru, and A. Jennings, “Energy efficient cluster formation in wireless sensor networks,” *10th International Conference on Telecommunications (ICT'03), vol. 2, Olynesia, France, February 2003*.

[46] A. Wang and A. P. Chandrakasan, “Energy-efficient DSPs for wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 19, no. 4 , pp. 68-78, July 2002.

[47] S. Mahlknecht and M. Böck, “CSMA-MPS: A minimum preamble sampling MAC protocol for low power wireless sensor networks,” in *Proceedings of the International Workshop on Factory Communication Systems, WFCS, pp. 73 - 80, 2004*.

[48] W. Ye, J. Heidermann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” USC/ISI Technical Report ISI-TR-543, Sep 2001.

[49] B. Krishnamachari, S. B. Wicker, and R. Bejar, "Phase transition phenomena in wireless ad-hoc networks," *GLOBECOM, San Antonio, TX, November 2001*.

[50] A. Varga, "OMNeT++ User Manual", ver. December 2004, http://www.omnetpp.org.

[51] "Mobility Framework for OMNeT++", http://mobility-fw.sourceforge.net.

CONTACT INFORMATION

E-mail address:	elarduserasmus@yahoo.co.uk
Mobile number:	+44-79-105-43546